

**UNIVERSITÉ BLAISE PASCAL - CLERMONT-FERRAND II**

*École Doctorale  
Sciences Pour l'Ingénieur de Clermont-Ferrand*

Thèse présentée par :  
**Vincent GAY-BELLILE**

Formation Doctorale CSTI :  
Composants et Systèmes pour le Traitement de l'Information

en vue de l'obtention du grade de

**DOCTEUR D'UNIVERSITÉ**

Spécialité : Vision par ordinateur

**Contributions au recalage et à la reconstruction 3D  
de surfaces déformables**

Soutenue publiquement le 10 novembre 2008 devant le jury :

M. Michel DHOME	Président
M. Nassir NAVAB	Rapporteur
M. Cristian SMINCHISESCU	Rapporteur
M. Vincent LEPETIT	Examineur
M. Etienne MÉMIN	Examineur
M. Adrien BARTOLI	Encadrant
M. Patrick SAYD	Co-encadrant
M. Jean-Thierry LAPRESTÉ	Directeur de thèse



# Résumé

Cette thèse porte sur le développement d'outils permettant le recalage d'images d'une surface déformable et la reconstruction tridimensionnelle de surfaces déformables à partir d'images prises par une seule caméra. Les surfaces que nous souhaitons traiter sont typiquement un visage ou une feuille de papier. Ces problématiques sont mal posées lorsque seule l'information présente dans les images est exploitée. Des informations *a priori* sur les déformations physiquement admissibles de la surface observée doivent être définies. Elles diffèrent en fonction du problème étudié. Par exemple, pour une feuille de papier, la courbure Gaussienne évaluée en chacun de ces points est nulle, cette propriété n'est pas valide pour un visage. Les applications visées sont l'insertion réaliste de logo 2D, de texte et également d'objets virtuels 3D dans des vidéos présentant une surface déformable.

La première partie de cette thèse est consacrée au recalage d'images par modèles déformables. Après avoir brièvement introduit les notions de base sur les fonctions de déformation et sur leur estimation à partir de données images, nous donnons deux contributions. La première est un algorithme de recalage d'images d'une surface déformable, qui est efficace en terme de temps de calcul. Nous proposons une paramétrisation par primitives des fonctions de déformation permettant alors leur estimation par des algorithmes compositionnels habituellement réservés aux transformations formant un groupe. La deuxième contribution est la modélisation explicite des auto-occultations, en imposant la contraction de la fonction de déformation le long de la frontière d'auto-occultation.

La deuxième partie de cette thèse aborde le problème de la reconstruction tridimensionnelle monoculaire de surfaces déformables. Nous nous basons sur le modèle de faible rang : les déformations sont approximées par une combinaison linéaire de modes de déformation inconnus. Nous supposons que ces derniers sont ordonnés par importance en terme d'amplitude de déformation capturée dans les images. Il en résulte une estimation hiérarchique des modes, facilitant l'emploi d'un modèle de caméra perspectif, la sélection automatique du nombre de modes et réduisant certaines ambiguïtés inhérentes au modèle. Nous explorons finalement la capture des déformations d'une surface peu texturée à partir de données issues d'un capteur 3D. L'information présente au niveau des contours de la surface est notamment utilisée.

Nous avons implantés les différentes contributions décrites ci-dessous. Elles sont testées et comparées à l'état de l'art sur des données réelles et synthétiques. Les résultats sont présentés tout au long du tapuscrit.

**Mots clés :** surfaces déformables, recalage d'images, reconstruction tridimensionnelle, "Structure-from Motion"



# Abstract

The goal of this thesis is to propose algorithms for the non-rigid image registration and 3D reconstruction of deformable surfaces from monocular videos. Sheets of paper or faces are typical surfaces we want to deal with. These problems are intrinsically ill-posed. Prior information has to be used. Their degree of specificity to the surface at hand may vary. For example, a paper is a surface with everywhere vanishing Gaussian curvature. This property can not be used for a face. These issues are important for tasks such as video augmentation by texture editing or by adding virtual 3D objects.

The first part of this thesis tackles non-rigid image registration. Previous work on warp-based deformation modeling and estimation methods are firstly described. We then present two contributions. The first one is an efficient registration algorithm using a feature-driven parametrization of the warp. This extends compositionnal algorithms to cope with non-rigid warps. Our second contribution is a specific framework for non-rigid image registration in spite of self-occlusions. The main idea is to force the warp to collapse at the self-occlusion boundary.

The second part of this thesis is concerned with 3D reconstruction of deformable surfaces. We use the low-rank shape model which represents the 3D shape as a linear combination of unknown shape bases. We propose a new way of looking at the low-rank shape model. Instead of considering it as a whole, we assume a coarse-to-fine ordering of the deformation modes. This allows us to propose a coarse-to-fine reconstruction algorithm which easily incorporates a perspective camera model, automatically selects the number of modes, and drastically reduces the level of ambiguity of the original low-rank shape model. Finally, we study deformation capture of untextured surfaces from 3D data using boundary information.

**Key-words :** deformable surfaces, image registration, 3D reconstruction, Structure-from-Motion



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Éléments de base</b>	<b>5</b>
1.1 Matrices, vecteurs et opérations associées . . . . .	5
1.2 Modèles de projection, coordonnées homogènes . . . . .	7
1.2.1 Coordonnées homogènes . . . . .	7
1.2.2 Modèle de projection perspectif . . . . .	7
1.2.3 Modèle de projection affine . . . . .	8
1.3 Mesures de distance . . . . .	9
1.4 Introduction à la reconstruction 3D de scènes rigides . . . . .	10
1.4.1 Initialisation à partir de deux vues . . . . .	10
1.4.1.1 La géométrie épipolaire . . . . .	10
1.4.1.2 Estimation des caméras et de la structure . . . . .	11
1.4.1.3 Normalisation des données . . . . .	11
1.4.2 Initialisation à partir de plusieurs vues . . . . .	11
1.4.3 Ajustement de faisceaux . . . . .	13
1.5 Images, transformations et augmentation . . . . .	14
1.5.1 Transformation entre images . . . . .	14
1.5.2 Augmentation 2D d'une image . . . . .	16
1.5.3 Augmentation 3D d'une image . . . . .	16
1.6 Optimisation numérique . . . . .	17
1.6.1 Moindres carrés linéaires . . . . .	18
1.6.2 Moindres carrés non-linéaires . . . . .	19
<b>2 Cadre général du recalage d'images par modèles déformables</b>	<b>21</b>
2.1 Introduction . . . . .	21
2.2 Cadre général de l'estimation . . . . .	22
2.2.1 Forme de la fonction de coût . . . . .	22
2.2.1.1 Critères directs . . . . .	23
2.2.1.2 Critères basés primitives . . . . .	24
2.2.1.3 Termes de régularisation . . . . .	24
2.2.1.4 Robustesse aux données aberrantes . . . . .	25
2.2.2 Méthodes directes . . . . .	26
2.2.2.1 Gestion des variations d'illumination . . . . .	26
2.2.2.2 Approche multi-résolution . . . . .	26
2.2.2.3 Méthodes d'optimisation . . . . .	27

2.2.3	Méthodes basées primitives . . . . .	32
2.3	Modélisation des déformations image . . . . .	34
2.3.1	Les fonctions à base radiale . . . . .	34
2.3.1.1	Définition . . . . .	34
2.3.1.2	Estimation des coefficients . . . . .	35
2.3.1.3	Propriétés des fonctions à base radiale . . . . .	35
2.3.1.4	Paramétrisation des fonctions à base radiale par primitives . . . . .	39
2.3.2	Déformations de forme libre . . . . .	40
2.3.2.1	Définition . . . . .	40
2.3.2.2	Interpolation par B-spline . . . . .	41
2.3.2.3	Illustration . . . . .	42
2.3.2.4	Paramétrisation des déformations de forme libre par primitives . . . . .	42
2.4	Modélisation des déformations 3D par une combinaison linéaire de modes . . . . .	43
2.4.1	Définitions des modèles de faible rang, « morphables » et actifs . . . . .	43
2.4.2	Quelques exemples de modèles pré-appris . . . . .	44
2.5	Conclusion . . . . .	48
<b>3</b>	<b>Algorithmes compositionnels pour modèles déformables</b>	<b>51</b>
3.1	Introduction . . . . .	51
3.2	Recalage compositionnel guidé par primitives . . . . .	53
3.2.1	Travaux antérieurs . . . . .	53
3.2.1.1	Approximation de la composition . . . . .	53
3.2.1.2	Approximation de l'inversion . . . . .	54
3.2.1.3	Discussion . . . . .	55
3.2.2	Le guidage par primitives . . . . .	55
3.2.3	Enchaînement des déformations . . . . .	56
3.2.3.1	Principe de base . . . . .	56
3.2.3.2	Estimation analytique pour les fonctions à base radiale . . . . .	57
3.2.4	Renversement d'une déformation . . . . .	58
3.2.4.1	Principe de base . . . . .	58
3.2.4.2	Estimation analytique pour les fonctions à base radiale . . . . .	58
3.2.5	Recalage compositionnel guidé par primitives . . . . .	59
3.3	Recalage basé apprentissage . . . . .	61
3.3.1	Introduction . . . . .	61
3.3.2	Approximation linéaire . . . . .	62
3.3.2.1	Apprendre une matrice d'interaction . . . . .	63
3.3.2.2	Limitations de l'approximation linéaire . . . . .	64
3.3.3	Approximation linéaire par morceaux . . . . .	65
3.3.3.1	Présentation . . . . .	65
3.3.3.2	Différentes relations linéaires par morceaux . . . . .	65
3.4	Résultats expérimentaux . . . . .	66
3.4.1	Comparaison des différentes méthodes d'apprentissage . . . . .	66
3.4.2	Comparaison entre les différentes approches existantes . . . . .	67
3.4.2.1	Données simulées . . . . .	69
3.4.2.2	Données réelles . . . . .	69



3.5	Conclusion . . . . .	75
<b>4</b>	<b>Gestion des auto-occultations en recalage d'images</b>	<b>77</b>
4.1	Introduction et travaux antérieurs . . . . .	77
4.2	Énoncé du problème et solution proposée . . . . .	79
4.2.1	Problématique . . . . .	79
4.2.2	Approche proposée . . . . .	79
4.2.3	Approche alternative . . . . .	81
4.3	Outils proposés pour la détection des auto-occultations . . . . .	82
4.3.1	Le contracteur . . . . .	82
4.3.2	Cartes d'auto-occultation binaires . . . . .	83
4.3.2.1	Approche basée sur la position . . . . .	83
4.3.2.2	Approche basée sur la dérivée . . . . .	83
4.3.2.3	Discussion sur la méthode de détection . . . . .	85
4.3.3	Carte d'auto-occultation probabiliste . . . . .	85
4.4	Recalage non-rigide avec auto-occultations . . . . .	86
4.4.1	Fonction de coût . . . . .	86
4.4.2	Occultations externes . . . . .	87
4.4.3	Optimisation par l'algorithme Gauss-Newton . . . . .	88
4.5	Augmentation 2D d'images d'une surface auto-occultée . . . . .	90
4.6	Résultats expérimentaux . . . . .	92
4.6.1	Le contracteur . . . . .	93
4.6.2	Une vidéo synthétique . . . . .	94
4.6.2.1	Description . . . . .	94
4.6.2.2	Résultats . . . . .	94
4.6.3	La vidéo de la feuille de papier . . . . .	96
4.6.3.1	Description de la vidéo . . . . .	96
4.6.3.2	La matrice Jacobienne . . . . .	96
4.6.3.3	Détection des auto-occultations . . . . .	97
4.6.3.4	Résultat du recalage et de l'augmentation 2D . . . . .	98
4.6.4	Comparaison avec les méthodes robustes . . . . .	98
4.6.5	Auto-occultation sur les bords de la surface . . . . .	100
4.6.6	Occultations externes . . . . .	101
4.6.7	La vidéo de la bande dessinée . . . . .	101
4.6.8	Autres types de surfaces . . . . .	102
4.6.9	Echecs . . . . .	102
4.7	Conclusion . . . . .	102
<b>5</b>	<b>Reconstruction 3D de surfaces déformables</b>	<b>105</b>
5.1	Introduction . . . . .	105
5.2	Approximation des déformations par combinaison linéaire de modes de déformation	107
5.2.1	Modèles pré-appris . . . . .	107
5.2.1.1	Les modèles morphables 3D . . . . .	108
5.2.1.2	Exemples de modèles morphables 3D . . . . .	108
5.2.2	Modèles non appris . . . . .	109

5.2.2.1	Le modèle de faible rang explicite . . . . .	109
5.2.2.2	Le modèle de faible rang implicite . . . . .	109
5.2.2.3	Les informations <i>a priori</i> . . . . .	110
5.3	Estimation de la forme explicite du modèle de faible rang . . . . .	111
5.3.1	L'approche stratifiée . . . . .	111
5.3.1.1	L'approche de Bregler <i>et al.</i> . . . . .	113
5.3.1.2	L'approche de Xiao <i>et al.</i> . . . . .	113
5.3.2	Estimation directe de la forme explicite . . . . .	114
5.3.3	Discussion sur les approches existantes . . . . .	116
5.4	Le modèle de faible rang hiérarchique . . . . .	117
5.4.1	Présentation générale . . . . .	117
5.4.2	Description du modèle de faible rang hiérarchique . . . . .	118
5.4.3	Estimation de la forme moyenne . . . . .	119
5.4.4	Triangulation d'un mode . . . . .	120
5.4.4.1	Initialisation des directions . . . . .	120
5.4.4.2	Initialisation des coefficients de forme et des amplitudes de défor- mation . . . . .	121
5.4.4.3	Raffinement non-linéaire . . . . .	122
5.4.5	Critère d'arrêt d'ajout de modes . . . . .	123
5.5	Résultats expérimentaux . . . . .	124
5.5.1	Données synthétiques . . . . .	124
5.5.2	Données réelles . . . . .	127
5.6	Conclusion . . . . .	132
<b>Conclusion</b>		<b>133</b>
<b>Annexes</b>		<b>135</b>
<b>A Capture de déformations à partir de données issues d'un capteur 3D</b>		<b>137</b>
A.1	Introduction . . . . .	137
A.2	Travaux antérieurs . . . . .	138
A.3	Énoncé du problème . . . . .	140
A.3.1	Représentation de la surface . . . . .	140
A.3.2	Fonction de coût . . . . .	140
A.4	Procédure de minimisation . . . . .	142
A.5	Résultats expérimentaux . . . . .	142
A.5.1	Une feuille de papier observée par un scanner à lumière structurée . . . . .	143
A.5.2	Une couverture observée par un système stéréoscopique . . . . .	144
A.5.3	Discussion . . . . .	146
A.6	Conclusion . . . . .	147
<b>Bibliographie</b>		<b>152</b>

# Table des figures

1.1	Le modèle de projection perspective ou sténopé. . . . .	8
1.2	Principe de l'augmentation 2D d'une image . . . . .	17
1.3	Construction de la carte d'occupation pour l'augmentation 2D d'une image . . . . .	18
2.1	Principe des méthodes directes de recalage d'images . . . . .	24
2.2	Exemples d'estimateurs robustes. . . . .	26
2.3	L'approche multi-résolution en recalage d'images . . . . .	27
2.4	Comparaison des lois de mises à jour additive et compositionnelle . . . . .	29
2.5	Principe des méthodes basées primitives de recalage d'images . . . . .	33
2.6	Exemples de fonctions noyau pour des transformations RBFs (1) . . . . .	37
2.7	Exemples de fonctions noyau pour des transformations RBFs (2) . . . . .	38
2.8	Exemples de fonctions noyau pour des transformations RBFs (3) . . . . .	38
2.9	Exemples de déformations par RBFs pour différentes fonctions noyau . . . . .	39
2.10	Exemples de RBFs guidées par primitives . . . . .	40
2.11	Base de polynômes pour des B-splines de degré 3. . . . .	42
2.12	Exemples de déformations par FFDs . . . . .	43
2.13	Exemples de modes de déformation pour un modèle actif de visage . . . . .	45
2.14	Illustration des ambiguïtés du modèle morphable 3D proposé par Blanz et Vetter . . . . .	45
2.15	Les quatre familles d'angles permettant de construire le modèle morphable proposé par Salzmann <i>et al.</i> . . . . .	46
2.16	Illustration des données d'apprentissage du modèle morphable proposé par Salzmann <i>et al.</i> . . . . .	46
2.17	Exemples de déformations introduites par le modèle morphable proposé par Salzmann <i>et al.</i> . . . . .	47
2.18	Exemples d'unités de forme du modèle <i>Candide-3</i> . . . . .	48
2.19	Exemples d'unités d'action du modèle <i>Candide-3</i> . . . . .	49
3.1	Composition approximative d'une transformation affine par morceaux . . . . .	54
3.2	Processus d'enchaînement des déformations. . . . .	56
3.3	Exemples d'enchaînement de déformations. . . . .	57
3.4	Processus de renversement d'une déformation. . . . .	59
3.5	Exemples de renversement d'une déformation. . . . .	60
3.6	Recalage compositionnel guidé par primitives. . . . .	62
3.7	Principe du recalage d'images basé apprentissage. . . . .	63
3.8	Limitation de l'approximation linéaire . . . . .	65

3.9	Comparaison des différentes relations linéaires par morceaux en terme de fréquence de convergence . . . . .	68
3.10	Comparaison des différentes relations linéaires par morceaux en terme de complexité de convergence et de précision . . . . .	68
3.11	Comparaison des algorithmes DA-GN, DA-ESM, IC-GN, DC-Le en terme de fréquence de convergence . . . . .	70
3.12	Comparaison des algorithmes DA-GN, DA-ESM, IC-GN, DC-Le en terme de précision	70
3.13	Comparaison des algorithmes DA-GN, DA-ESM, IC-GN, DC-Le en terme de complexité de convergence . . . . .	71
3.14	Déformations estimées par l'algorithme DC-Le sur la vidéo du Tshirt . . . . .	72
3.15	Résultat du recalage et de l'augmentation 2D pour l'algorithme DC-Le sur la vidéo de la feuille de papier . . . . .	73
3.16	Déformations estimées par l'algorithme DC-Le sur la vidéo du tapis . . . . .	74
3.17	Déformations estimées par l'algorithme DC-Le sur la deuxième vidéo du Tshirt . . .	74
4.1	Comportements d'une fonction de déformation le long d'une frontière d'auto-occultation	80
4.2	Fonction de coût classique et celle modélisant explicitement les auto-occultations . .	81
4.3	Les composantes du contracteur . . . . .	82
4.4	Principe du module de détection des auto-occultations basé position . . . . .	84
4.5	Principe du module de détection des auto-occultations basé dérivée . . . . .	85
4.6	Propriétés de la fonction sigmoïde utilisée . . . . .	86
4.7	Cartes d'auto-occultation binaire et probabiliste . . . . .	87
4.8	Structures des matrices Jacobienne et Hessienne . . . . .	90
4.9	Contraintes temporelles sur la trajectoire des points de contrôle . . . . .	91
4.10	Illustration du processus d'augmentation 2D d'images d'une surface auto-occultée .	92
4.11	Exemples de recalage d'images avec ou sans contracteur . . . . .	94
4.12	Vidéo synthétique d'une feuille de papier auto-occultée . . . . .	95
4.13	Résultat du recalage et de la détection des auto-occultations sur la vidéo synthétique .	96
4.14	La vidéo de la feuille de papier . . . . .	96
4.15	Détection des pixels auto-occultés sur la vidéo de la feuille de papier . . . . .	97
4.16	Résultat du recalage et de l'augmentation 2D sur la vidéo de la feuille de papier . . .	98
4.17	Comparaison de l'approche proposée avec les méthodes robustes sur la deuxième vidéo de la feuille de papier . . . . .	99
4.18	Résultat du recalage sur des vidéos présentant des surfaces auto-occultées sur les bords	100
4.19	Erreurs résiduelles (RMS) en terme d'intensité pour différentes vidéos . . . . .	101
4.20	Résultat du recalage sur une vidéo présentant une occultation externe et une auto-occultation . . . . .	102
4.21	Augmentation 2D de la vidéo bande dessinée . . . . .	103
4.22	Résultat du recalage sur la vidéo du tapis . . . . .	104
4.23	Augmentation 2D de la vidéo du tissu . . . . .	104
4.24	Déformations faisant échouer l'approche proposée . . . . .	104
5.1	Exemples d'informations <i>a priori</i> utilisées dans la littérature pour la reconstruction 3D d'objets déformables . . . . .	106
5.2	Factorisation de la matrice de mesure en une matrice de mouvement et une matrice de forme . . . . .	112

5.3	Structures des matrices Jacobienne et Hessienne sur un exemple synthétique constitué de $n = 4$ vues et $m = 5$ points. . . . .	123
5.4	Comparaison sur des données de synthèse d'algorithmes de reconstruction 3D basés sur le modèle de faible rang . . . . .	126
5.5	La vidéo du visage. . . . .	128
5.6	Résultat de la reconstruction sur la vidéo du visage. . . . .	128
5.7	Résultat de la reconstruction sur la vidéo de la feuille de papier. . . . .	129
5.8	Résultat de la reconstruction sur la vidéo de la feuille de papier en présence de données manquantes. . . . .	129
5.9	La vidéo "Desperate Housewives". . . . .	130
5.10	Résultat de la reconstruction sur la vidéo "Desperates Housewives" . . . . .	131
5.11	Erreur de reprojection et score de validation croisée sur la vidéo "Desperates Housewives" . . . . .	131
5.12	La vidéo de la feuille de papier auto-occultée et les données associées . . . . .	131
5.13	Résultat de la reconstruction sur la vidéo de la feuille de papier auto-occultée . . . . .	132
A.1	Transformée en distance sur un nuage de points 2D . . . . .	143
A.2	Structures des matrices Jacobienne et Hessienne . . . . .	144
A.3	Résultat du recalage sur trois nuages de points 3D provenant d'un scanner à lumière structurée . . . . .	145
A.4	Images synthétiques générées avec différentes textures. . . . .	146
A.5	Images synthétisées en interpolant les surfaces reconstruites à partir des données des figures A.3(a), A.3(b) et A.3(c). . . . .	148
A.6	La séquence de la couverture . . . . .	149
A.7	Extraction des données d'entrées pour la séquence de la couverture . . . . .	149
A.8	Résultat du recalage sur la séquence de la couverture . . . . .	150
A.9	Vidéo de synthèse représentant un film sur un écran déformable . . . . .	151



# Liste des tableaux

2.1	Les différentes terminologies utilisées pour nommer les algorithmes minimisant le critère SSD. . . . .	29
2.2	Les différentes étapes d'une itération de l'algorithme DA-GN. . . . .	30
2.3	Les différentes étapes d'une itération de l'algorithme DA-ESM. . . . .	31
2.4	Les différentes étapes d'une itération de l'algorithme IC-GN. . . . .	31
2.5	Récapitulation des modèles représentant les déformations d'une surface par une combinaison linéaire de modes de déformation . . . . .	44
2.6	Les AUs du modèle <i>Candide-3</i> . . . . .	48
2.7	Les SUs du modèle <i>Candide-3</i> . . . . .	48
3.1	Les différentes étapes d'une itération de l'algorithme IC-GN guidé par primitives . . . . .	61
3.2	Les différentes étapes d'une itération de l'algorithme DC-Le guidé par primitives . . . . .	63
3.3	Les bornes de déplacement utilisées pour apprendre les différentes matrices d'interaction . . . . .	66
3.4	Résultat sur la vidéo du Tshirt . . . . .	71
3.5	Résultat sur la vidéo de la feuille de papier . . . . .	72
3.6	Résultat sur la vidéo du tapis . . . . .	72
3.7	Résultat sur la deuxième vidéo du Tshirt . . . . .	74
4.1	Les différentes étapes de notre algorithme de recalage d'images d'une surface déformable en présence d'auto-occultation . . . . .	89
5.1	Comparaison des différentes méthodes de reconstruction 3D basées sur le modèle de faible rang . . . . .	116
5.2	Principe de l'estimation hiérarchique du modèle de faible rang . . . . .	119
5.3	Les différentes étapes de l'estimation hiérarchique du modèle de faible rang. . . . .	125
A.1	Temps de calcul requis pour reconstruire les déformations d'une surface avec l'algorithme proposé . . . . .	147





# Introduction

L'image est omniprésente dans nos sociétés modernes principalement, en raison de la production grandissante d'ordinateurs puissants et de capteurs visuels à bas coût. Les exemples les plus frappants sont les appareils photos numériques, les webcams et les caméraphones qui envahissent notre quotidien. Ils fournissent des images et des vidéos de qualités variées. Le développement de tous ces appareils entraîne une forte demande d'algorithmes et logiciels permettant de traiter ces images et flux vidéos. Par conséquent, la recherche dans le domaine de la vision par ordinateur est en plein développement.

Le principe de la vision par ordinateur est de permettre à une machine d'interagir avec l'environnement observé par une ou plusieurs caméras. Il est pour cela nécessaire d'extraire de l'information à partir des images ou des vidéos générées. Les applications possibles sont nombreuses et dans des domaines variés : surveillance, métrologie, médical, robotique, *etc.*

La plupart des problèmes de vision par ordinateur requièrent une étape de modélisation et une étape d'estimation. L'étape de modélisation consiste à formuler un modèle mathématique décrivant le problème et les contraintes qui lui sont associées, ainsi qu'une fonction de coût dont le minimum correspond à la solution recherchée. Le processus d'estimation revient à calculer les paramètres du modèle à partir d'observations en minimisant la fonction de coût. Même si ce principe est général, son application à l'étude de surfaces déformables est en plein développement que depuis peu de temps, en raison des difficultés à modéliser les déformations et à estimer les modèles complexes introduits. Or notre environnement est par nature non-rigide, car il est en partie constitué d'objets en mouvement et déformables, tels les matériaux souples (tissus, caoutchouc), les personnes ou les plantes. Pour cette raison, les techniques de vision par ordinateur adaptées aux scènes non-rigides présentent de nombreuses applications potentielles.

Cette thèse s'inscrit dans ce courant scientifique et s'appuie sur les développements récents de modèles et d'algorithmes spécifiques aux objets déformables. Nous nous intéressons plus particulièrement à deux problématiques majeures : le recalage d'images d'une surface déformable et la reconstruction tridimensionnelle des déformations d'une surface à partir d'images prises par une seule caméra.

Le principe du recalage d'images est d'estimer les transformations géométrique et photométrique qui rendent deux ou plusieurs images les plus « semblables » possible. Dans la littérature, il existe deux grandes approches au recalage d'images. L'approche dite directe, utilise l'information de couleur contenue par chaque pixel des images. Un critère possible est la différence de couleur entre pixels correspondants. Les approches basées primitives s'appuient sur un critère de distance entre primitives mises en correspondances (points, droites, *etc.*). De manière générale, le recalage non-rigide requiert une fonction de coût constituée de deux termes : un terme de données (direct ou basé primitives)

rassemblant les observations image et un terme de régularisation. Sans ce dernier, le problème est sous-contraint. Le résultat du processus de recalage d'images est un champ de déplacement dense entre les différentes images, pouvant être utilisé par exemple pour l'insertion d'une texture en réalité augmentée, et pour la fusion multimodale d'images en imagerie médicale.

Le principe de la reconstruction tridimensionnelle est d'estimer la structure 3D de la scène ainsi que le déplacement de la caméra. Pour une scène rigide, les rayons de vue associés au même point 3D observé par différentes caméras s'intersectent dans l'espace. Cette propriété permet de définir une contrainte forte sur la reconstruction. La reconstruction tridimensionnelle de scènes rigides est en général un problème bien posé. En revanche, cette propriété est perdue lorsque la surface observée se déforme : la reconstruction tridimensionnelle monoculaire de surfaces déformables est un problème intrinsèquement mal posé. Il est en général indispensable d'incorporer des contraintes supplémentaires pour pouvoir le résoudre. La grande difficulté revient à correctement modéliser le problème en contraignant suffisamment la reconstruction tout en ne pénalisant pas les déformations physiquement plausibles de la surface observée. Les applications inhérentes à la reconstruction 3D de surfaces déformables sont nombreuses dans l'industrie du film pour les effets spéciaux, en métrologie pour la mesure de déformations sous contraintes mécaniques et pour la modélisation d'objets souples : tissus vivants, textiles *etc.*

**Contexte.** Cette thèse a été effectuée en collaboration entre le LASMEA UMR 6602 du CNRS de l'Université Blaise Pascal à Clermont-Ferrand et le LSVE du CEA LIST à Saclay. Les deux premières années de la thèse ont été accomplies au LASMEA et la dernière année au LSVE. Les travaux présentés sont également issus de collaborations entretenues lors de stages à l'étranger. Ainsi, j'ai passé un mois au laboratoire IPAB à Edimbourg sous la direction du Professeur Bob Fisher, deux semaines au laboratoire VIPS à Vérone sous la direction d'Umberto Castellani et enfin deux mois à l'Université Queen Mary à Londres sous la direction de Lourdes Agapito.

**Contributions.** Nos contributions se situent aussi bien au niveau de la modélisation des déformations que du point de vue de leur estimation à partir d'observations image. En recalage d'images, nous avons proposé un algorithme permettant d'estimer efficacement les déformations ainsi qu'une modélisation explicite des auto-occlusions. Pour la reconstruction tridimensionnelle, nous avons proposé une nouvelle modélisation des déformations 3D via une combinaison linéaire de modes de déformation ordonnés, pour laquelle nous avons introduit une estimation hiérarchique des modes. Nous listons plus en détails ci-dessous les différentes contributions apportées dans ce tapuscrit.

- ▷ *Recalages non-rigides guidés par primitives (chapitre 3).* Nous proposons le concept de recalage d'images guidé par primitives, dont le principe est de paramétrer la fonction de déformation par un ensemble de primitives. Nous montrons que cette paramétrisation permet d'étendre les algorithmes compositionnels aux modèles déformables. Ces approches sont plus efficaces en terme de temps de calcul que les algorithmes « classiques » à mise à jour additive. La contrainte de groupe sur la transformation géométrique inhérente aux algorithmes compositionnels est relaxée en opérant directement sur les primitives pour réaliser des opérations telles que le renversement et l'enchaînement de déformations, approximant respectivement l'inversion et la composition de fonctions. Celles-ci n'existent pas pour des transformations ne formant pas un groupe. Nous montrons que les algorithmes compositionnels inverses et basés sur un apprentissage hors ligne peuvent être utilisés avec des fonctions de déformation de type

plaque mince.

- ▷ *Modélisation explicite des auto-occultations pour le recalage d'images (chapitre 4).* Au cours de nos travaux, nous nous sommes intéressés aux problèmes engendrés par les auto-occultations en recalage d'images. Les travaux précédents avaient jusqu'alors privilégié une fonction de coût invariante aux effets introduits par les auto-occultations en incluant un estimateur robuste dans le terme de données. Nous modélisons explicitement les auto-occultations en 2D pour le recalage d'images. Nous imposons à la fonction de déformation de se comprimer le long de la frontière d'auto-occultation en ajoutant un terme appelé le contracteur à la fonction de coût classique. Cette propriété permet d'utiliser le comportement de la fonction de déformation pour détecter les régions auto-occultées. Nous proposons un opérateur de détection se basant sur le principe que la contraction introduit une dérivée nulle de la fonction de déformation dans au moins une direction.
- ▷ *Le modèle de faible rang hiérarchique (chapitre 5).* Nous proposons un nouvel algorithme de reconstruction tridimensionnelle basé sur le modèle de faible rang : les déformations de la surface sont représentées par une combinaison linéaire de modes de déformation. Nous proposons une nouvelle représentation de ce modèle en considérant que les modes sont ordonnés par importance en terme d'amplitude de déformation capturée dans les images. Cette représentation présente de nombreux avantages par rapport aux méthodes qui utilisent le modèle de faible rang comme un ensemble de modes non-ordonnés. Il en résulte une estimation hiérarchique des modes facilitant l'emploi d'un modèle de caméra perspectif et la sélection automatique du nombre de modes. Le degré d'ambiguïté lié au modèle est en outre fortement réduit.
- ▷ *Capture de déformations à partir de données issues d'un capteur 3D (Annexe A).* Nous explorons finalement la capture des déformations d'une surface peu texturée à partir de données issues d'un capteur 3D. L'algorithme proposé déforme un maillage régulier jusqu'à ce qu'il corresponde « au mieux » au nuage de points 3D. Pour cela une fonction de coût constituée de plusieurs termes de données et de régularisation est minimisée. L'algorithme proposé est également rendu robuste afin de gérer la présence de données manquantes et / ou erronées.

**Plan.** Nous avons donné ci-dessus une introduction générale aux problèmes étudiés dans cette thèse. Le chapitre 1 présente les éléments de bases requis tout au long du tapuscrit : les différentes notations, les notions de projection et de distance, une introduction à la reconstruction 3D de scène rigide, les méthodes d'optimisation, entre autres. Le chapitre 2 décrit le cadre général du recalage d'images par modèles déformables. Les fonctions de déformation existantes ainsi que les méthodes permettant de les estimer à partir d'informations image sont présentées. Dans le chapitre 3, nous cherchons à améliorer l'efficacité en terme de temps de calcul des approches directes. Le chapitre 4 aborde la problématique des auto-occultations en recalage d'images. Au chapitre 5, nous étudions la reconstruction tridimensionnelle de surfaces déformables par un modèle de faible rang. Nous concluons et discutons des perspectives avant de terminer le tapuscrit par l'annexe A sur la capture de déformations à partir de données issues d'un capteur 3D.



# Chapitre 1

## Éléments de base

---

---

Ce chapitre expose les éléments de base nécessaires à la lecture de cette thèse. Nous y définissons les conventions et notations utilisées et y développons l'emploi de certains opérateurs et décompositions de matrices. Nous décrivons des mesures de distance Euclidiennes et algébriques et présentons les modèles de caméra utilisés. Nous introduisons le principe de reconstruction 3D d'une scène rigide, ainsi que les concepts d'augmentation 2D et 3D d'image. Pour finir nous exposons différentes méthodes d'optimisation au sens des moindres carrés.

---

---

### 1.1 Matrices, vecteurs et opérations associées

Les matrices sont notées en utilisant la police *sans-serif* ( $A$ ) ou calligraphique ( $\mathcal{A}$ ). Les vecteurs sont représentés en gras ( $\mathbf{a}$ ,  $\mathbf{A}$ ). La transposée, l'inverse et la pseudo-inverse d'une matrice sont respectivement notées  $A^T$ ,  $A^{-1}$  et  $A^\dagger$ . Cette dernière est généralement<sup>1</sup> définie par  $A^\dagger \stackrel{\text{def}}{=} (A^T A)^{-1} A^T$ . La norme deux d'un vecteur est définie par :  $\|\mathbf{v}\|^2 \stackrel{\text{def}}{=} \sum_{i=1}^p v_i^2$ , avec  $v_i$  le  $i^{\text{ème}}$  élément du vecteur  $\mathbf{v}$  et  $p$  le nombre d'éléments le constituant. Le RMS<sup>2</sup>  $\text{rms}(\mathbf{v}) = \sqrt{\frac{1}{p} \|\mathbf{v}\|^2}$  est proportionnel à la norme deux.

**Opérations sur les matrices.** L'opérateur "vect" vectorise une matrice par ligne. Par exemple si les  $p$  lignes de la matrice  $A$  sont notés  $\mathbf{r}_i^T$  :

$$\text{vect}(A) \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{r}_1 \\ \vdots \\ \mathbf{r}_p \end{pmatrix} \quad \text{avec} \quad A = \begin{pmatrix} \mathbf{r}_1^T \\ \vdots \\ \mathbf{r}_p^T \end{pmatrix}.$$

L'opérateur "diag" appliqué à un vecteur  $\mathbf{v}$  introduit une matrice carrée dont les éléments de la diagonale sont les composantes de  $\mathbf{v}$  :

$$\text{diag}(\mathbf{v}) \stackrel{\text{def}}{=} \begin{pmatrix} v_1 & & \\ & \ddots & \\ & & v_p \end{pmatrix}.$$

---

<sup>1</sup>Notons que la pseudo inverse de la matrice  $A$  peut être définie même si la matrice  $(A^T A)$  n'est pas inversible.

<sup>2</sup>"Root mean square" en anglais.

Il représente également l'opération de concaténation diagonale par bloc lorsqu'il s'applique à plusieurs matrices :

$$\text{diag}(A, B) \stackrel{\text{def}}{=} \begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}.$$

La multiplication de matrices élément par élément, également appelée *produit d'Hadamard*, est symbolisée par  $\odot$ . L'opérateur "et" logique est noté  $\&$ . Il s'applique élément par élément sur des matrices binaires.

**Matrices fréquemment utilisées.** La matrice identité est notée  $I$ , la matrice  $\mathcal{S}$  est définie par

$$\mathcal{S} \stackrel{\text{def}}{=} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

La matrice  $[\mathbf{v}]_{\times}$  est définie telle que pour tous vecteurs  $\mathbf{v}$  et  $\mathbf{v}'$  de dimension 3,  $[\mathbf{v}]_{\times} \mathbf{v}' \stackrel{\text{def}}{=} \mathbf{v} \times \mathbf{v}'$ , où  $\times$  symbolise le produit vectoriel entre deux vecteurs. Cette matrice est antisymétrique et de rang 2. Elle est donnée par :

$$[\mathbf{v}]_{\times} = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}.$$

**Notions de semi-groupe et de groupe.** Un *semi groupe* est un couple  $(\mathcal{G}, \circ)$  où  $\mathcal{G}$  est un ensemble muni d'une loi de composition interne  $\circ$  tel que le couple  $(\mathcal{G}, \circ)$  possède les propriétés suivantes  $\forall a, b, c \in \mathcal{G}$  :

▷ **Existence d'un élément neutre :**  $\exists e \in \mathcal{G}, a \circ e = e \circ a = a$ .

▷ **Régularité des éléments :** si  $a \circ b = a \circ c$  ou  $b \circ a = c \circ a$ , alors  $b = c$ .

▷ **Associativité de la loi :**  $(a \circ b) \circ c = a \circ (b \circ c)$ .

Pour un *groupe* la propriété supplémentaire d'existence d'un inverse pour tous les éléments de  $\mathcal{G}$  est requise :  $a \circ b = b \circ a = e$ .  $b$  est dit symétrique de  $a$  et on le note  $a^{-1}$ .

Nous notons par la suite  $O(r)$  le groupe des matrices orthogonales réelles de taille  $r \times r$ . Elles vérifient  $AA^T = A^T A = I$ . De même, nous notons  $SO(r)$  le groupe des matrices orthonormales directes de taille  $(r \times r)$ . Elles vérifient :  $AA^T = A^T A = I$  et  $\det(A) = 1$ .

**Décomposition en valeurs singulières.** (SVD pour "Singular Value Decomposition" en anglais). Cet outil permet d'obtenir la solution d'un système linéaire exact, la solution au sens des *moindres carrés* d'un système linéaire sur-contraint, et donne une famille de solutions pour un système linéaire sous contraint. La SVD est basée sur le théorème suivant : toute matrice  $A$  de taille  $(r \times c)$  peut être décomposée comme suit :

$$A_{(r \times c)} = \Xi_{r \times c} \Sigma_{c \times c} \Gamma_{c \times c}^T,$$

où  $\Xi$  est une matrice orthonormale par colonne<sup>3</sup> et  $\Gamma \in SO(c)$ . Les matrices  $\Xi$  et  $\Gamma$  contiennent les vecteurs singuliers de  $A$ .  $\Sigma$  est une matrice diagonale contenant les valeurs singulières de  $A$ . La

<sup>3</sup> $\Xi^T \Xi = I$  (mais pas nécessairement  $\Xi \Xi^T = I$ ).

plupart des algorithmes de décomposition en valeurs singulières ordonnent ces dernières de manière décroissante, nous supposons par la suite que cet ordonnancement est réalisé. Lorsque le rang de  $A$  n'est pas plein, les dernières colonnes de  $\Gamma$  forment une base orthonormale du noyau de  $A$ .

Notons que la SVD permet l'analyse en composantes principales (ACP) d'un ensemble de données. Elle consiste à rechercher les directions de l'espace qui représentent au mieux les corrélations entre des variables aléatoires. Ces directions sont obtenues par les colonnes de  $\Gamma$ . L'analyse des valeurs singulières permet de sélectionner les directions les plus pertinentes, également appelées axes principaux. En effet, plus une valeur singulière est élevée et plus la direction associée donnée par le vecteur singulier correspondant est représentative des données. L'ACP est notamment utilisée pour construire des modèles de déformation linéaires. De plus amples détails sont donnés en §2.4.

## 1.2 Modèles de projection, coordonnées homogènes

Nous introduisons brièvement dans cette section les notions de projection et de *coordonnées homogènes*. Pour plus de détails, se rapporter à (Faugeras, 1993; Hartley and Zisserman, 2003). Ces notions sont principalement utilisées au chapitre 5.

### 1.2.1 Coordonnées homogènes

Les points des espaces projectifs  $\mathbb{P}^2$  et  $\mathbb{P}^3$  sont représentés par des coordonnées homogènes et respectivement notés  $\tilde{\mathbf{q}} = \begin{pmatrix} x & y & w \end{pmatrix}$  et  $\tilde{\mathbf{Q}} = \begin{pmatrix} X & Y & Z & W \end{pmatrix}$ . Par la suite, la présence et l'absence d'un tilde sur une entité signifient respectivement que celle-ci est n'est pas exprimée en coordonnées homogènes. Deux vecteurs  $\tilde{\mathbf{v}}$  et  $\tilde{\mathbf{v}}'$  de coordonnées proportionnelles correspondent au même point de l'espace projectif : on note  $\tilde{\mathbf{v}} \sim \tilde{\mathbf{v}}'$ , où  $\sim$  représente l'égalité à un facteur près. Les coordonnées homogènes permettent de caractériser le plan à l'infini comme l'ensemble des points ou vecteurs qui ont leur dernière coordonnée nulle ( $w = 0$  ou  $W = 0$ ). Une transformation de l'espace projectif  $\tilde{A} : \mathbb{P}^p \rightarrow \mathbb{P}^d$  peut être représentée par une matrice  $(d+1) \times (p+1)$  en coordonnées homogènes. L'opération de projection, quelque soit le modèle de caméra utilisé (perspectif ou affine), est notée  $\Pi$ .

### 1.2.2 Modèle de projection perspectif

Ce modèle de caméra, appelé également modèle sténopé, modélise la projection perspective inhérente à la plupart des caméras. Il est illustré par la figure 1.1. Soit  $\tilde{\mathbf{Q}}$  un point de l'espace projectif  $\mathbb{P}^3$ . Sa projection, le point  $\tilde{\mathbf{q}}$ , est définie par l'intersection de la *rétine* avec le *rayon de projection*. Ce dernier est formé par le *centre de projection* et le point  $\tilde{\mathbf{Q}}$ . La projection est modélisée par une transformation projective de  $\mathbb{P}^3$  vers  $\mathbb{P}^2$ , représentée par une matrice  $\tilde{P}$  de taille  $(3 \times 4)$ . L'équation de projection est :

$$\tilde{\mathbf{q}} \sim \tilde{P}\tilde{\mathbf{Q}}.$$

Avec  $\tilde{P} \stackrel{\text{def}}{=} K \begin{pmatrix} R & \mathbf{t} \end{pmatrix}$ . La matrice  $K$  contient les *paramètres intrinsèques* ou internes de la caméra, tandis que le couple  $\{R, \mathbf{t}\}$  représente ses *paramètres extrinsèques*. Ils définissent la position et l'orientation de la caméra par une transformation rigide entre le repère caméra et le repère monde, c'est-à-dire le repère dans lequel sont exprimées les coordonnées des points 3D.

Les paramètres intrinsèques sont les caractéristiques internes de la caméra, liés notamment à la taille des pixels, au zoom, *etc.* La matrice  $K$  est donnée par :

$$K \stackrel{\text{def}}{=} \begin{pmatrix} \kappa f_o & \tau & u_0 \\ 0 & f_o & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Elle est triangulaire supérieure. Les paramètres internes sont :

- ▷ La *distance focale*  $f_o$ , distance orthogonale du centre optique au plan image exprimée en pixel par unité métrique.
- ▷ Le *point principal*  $(u_0 \ v_0)^T$  qui est l'intersection de l'axe optique et du plan image.
- ▷  $\kappa$  est le *rapport d'aspect* entre les dimensions verticale et horizontale des pixels du capteur photosensible.
- ▷  $\tau$  est un *facteur d'obliquité* lié à l'angle entre les axes du capteur.

En pratique, on fixe  $\tau = 0$  et  $\kappa = 1$ , ce qui est une bonne approximation des caméras réelles modernes. L'estimation des paramètres internes d'une caméra constitue l'opération de *calibrage*.

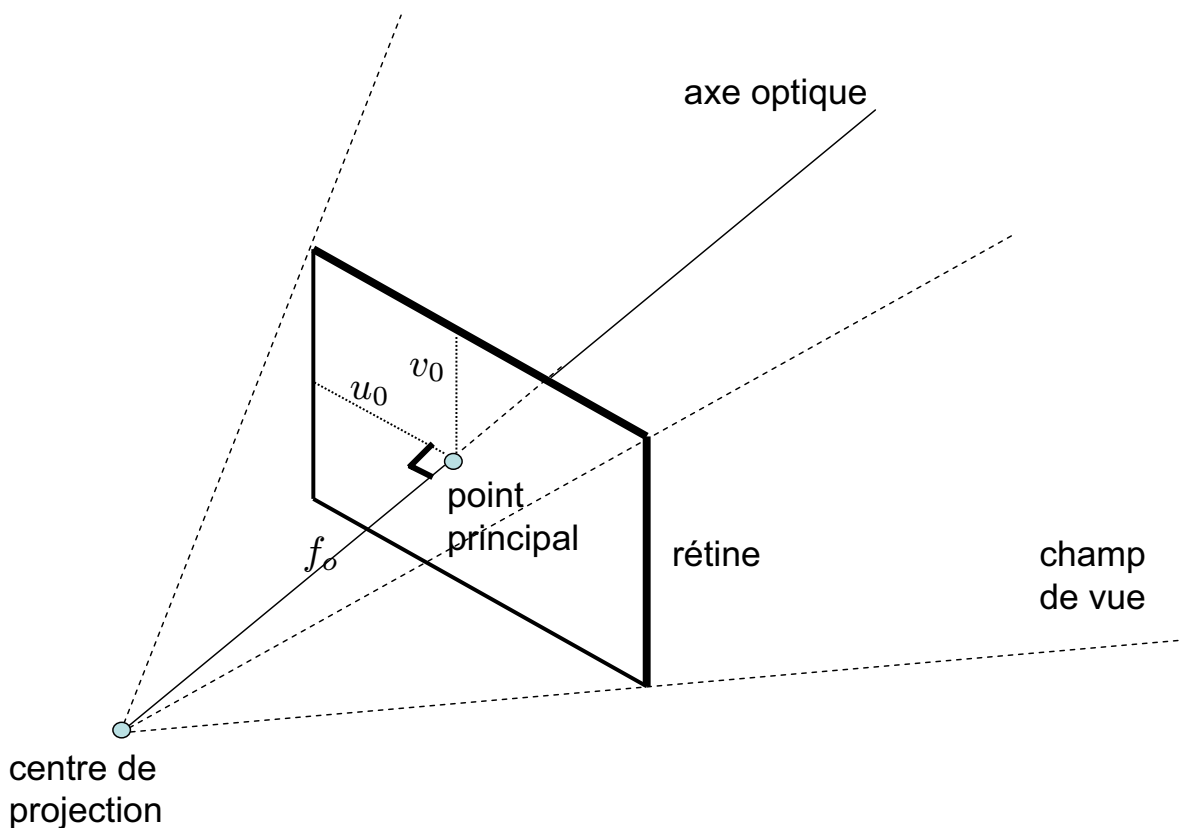


FIG. 1.1 – Le modèle de projection perspective ou « sténopé ».

### 1.2.3 Modèle de projection affine

Ce modèle décrit les caméras réalisant une projection pour laquelle les rayons de vue sont parallèles. La dernière ligne de la matrice de projection est donnée par  $p_3^T = (0 \ 0 \ 0 \ 1)$ . Un point à



l'infini est alors projeté à l'infini. Le modèle de caméra affine est décrit par :

$$\tilde{\mathbf{P}}_A \stackrel{\text{def}}{=} \begin{pmatrix} \alpha_x & \tau & 0 \\ 0 & \alpha_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \bar{\mathbf{R}}_{2 \times 3} & \bar{\mathbf{t}}_{2 \times 1} \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix}, \quad (1.1)$$

avec  $\bar{\mathbf{R}}$  la matrice de rotation tronquée aux deux premières lignes. Les paramètres internes sont le facteur d'obliquité  $\tau$  et les facteurs d'échelles suivant  $x$  et  $y$  :  $\alpha_x$  et  $\alpha_y$  respectivement.

La projection par caméra affine peut être vue comme une transformation affine d'un point  $\mathbf{Q}$  en coordonnée non-homogène :

$$\mathbf{q} = \mathcal{A}\mathbf{Q} + \mathbf{t}_A \quad \text{avec} \quad \mathcal{A} = \begin{pmatrix} \alpha_x & \tau \\ 0 & \alpha_y \end{pmatrix} \bar{\mathbf{R}} \quad \text{et} \quad \mathbf{t}_A = \begin{pmatrix} \alpha_x & \tau \\ 0 & \alpha_y \end{pmatrix} \bar{\mathbf{t}}.$$

L'une des principales propriétés des caméras affines est la conservation du parallélisme. Deux droites parallèles dans l'espace 3D sont projetées en deux droites parallèles dans les images. Ce modèle constitue une bonne approximation des caméras réelles si la scène observée présente de faibles variations de profondeur et si la distance focale est grande.

### 1.3 Mesures de distance

Dans notre étude, nous avons à estimer les paramètres d'un modèle à partir de critères basés sur des distances entre données image. Un critère possible est de minimiser une somme de distances au carré entre les points extraits d'une image et ceux prédits par un modèle. Des distances *algébriques* et *Euclidiennes* sont utilisées. La première l'est principalement pour obtenir une estimation du vecteur de paramètres inhérent au modèle, par résolution d'un problème de moindres carrés linéaires. Cette estimation est ensuite raffinée par une minimisation non-linéaire de distances Euclidiennes. La distance algébrique n'a pas de sens strictement physique contrairement à la distance Euclidienne.

**Distance Euclidienne.** Nous introduisons les distances Euclidiennes entre deux points et entre un point et une droite. Elles sont exprimées en terme des coordonnées homogènes de ces entités.

- ▷ **Distance point-point :** Elle est notée  $d_{pp}(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}')$  avec  $\mathbf{q}$  et  $\mathbf{q}'$  deux points de l'espace Euclidien de dimension  $p$  ( $p = 2$  ou  $p = 3$ ) représentés par leurs coordonnées homogènes :  $\tilde{\mathbf{q}}^T \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{q}^T & w \end{pmatrix}$  et  $\tilde{\mathbf{q}}'^T \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{q}'^T & w' \end{pmatrix}$ . Cette distance est donnée par :

$$d_{pp}^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}') \stackrel{\text{def}}{=} \left\| \frac{\mathbf{q}}{w} - \frac{\mathbf{q}'}{w'} \right\|^2.$$

Afin de ne pas alourdir les notations elle est également notée  $d^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}')$  dans le tapuscrit.

- ▷ **Distance point-droite :** La distance entre un point  $\mathbf{q}$  et une droite  $\mathbf{l}$  représentés par leurs coordonnées homogènes :  $\tilde{\mathbf{q}}^T \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{q}^T & w \end{pmatrix}$  et  $\tilde{\mathbf{l}}^T \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{l}^T & l \end{pmatrix}$  est donnée par :

$$d_{pl}^2(\tilde{\mathbf{q}}, \tilde{\mathbf{l}}) \stackrel{\text{def}}{=} \frac{(\mathbf{q}^T \mathbf{l})^2}{w^2 \|\mathbf{l}\|^2}.$$

**Distance algébrique.** Nous introduisons une distance algébrique entre deux points et entre un point et une droite. Ces distances sont également exprimées en terme de coordonnées homogènes.

- ▷ **Distance point-point :** Une distance algébrique entre deux points appartenant au plan Euclidien ( $p = 2$ ) est donnée par :

$$d_{A\_pp}^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}') \stackrel{\text{def}}{=} \left\| \mathcal{S}[\tilde{\mathbf{q}}]_{\times} \tilde{\mathbf{q}}' \right\|^2,$$

également notée  $d_A^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}')$  par la suite pour simplifier les notations.

- ▷ **Distance point-droite :** La distance algébrique entre un point et une droite s'écrit :

$$d_{A\_pl}^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}') \stackrel{\text{def}}{=} \left\| \tilde{\mathbf{q}}^T \tilde{\mathbf{l}} \right\|^2.$$

**Transformée en distance.** Elle associe à chaque point  $\mathbf{q}$  de l'espace la distance au point étiqueté  $\mathbf{q}^l$  le plus proche. Il en découle une carte de distance  $\mathcal{H}_D$  avec  $\mathcal{H}_D(\mathbf{q}) = d^2(\tilde{\mathbf{q}}, \tilde{\mathbf{q}}^l)$ . Cette opération est notamment utilisée sur des images pour des opérations morphologiques telles que l'*érosion* ou la *dilatation*. Plus de détails sur le calcul de la transformée en distance peuvent être trouvés dans (Borgefors, 1988). Nous utilisons la transformée en distance en annexe A.

## 1.4 Introduction à la reconstruction 3D de scènes rigides

Nous introduisons brièvement quelques techniques de reconstruction 3D de scènes rigides, des détails supplémentaires sont disponibles dans (Hartley and Zisserman, 2003). La plupart des algorithmes suivent un schéma d'estimation constitué de deux étapes :

- ▷ **Initialisation :** une estimation initiale des caméras et des points 3D est obtenue par minimisation d'un critère basé sur des distances algébriques.
- ▷ **L'ajustement de faisceaux :** les positions des points 3D et les poses des caméras sont raffinées par minimisation de distances Euclidiennes.

Les notions de base présentées dans cette section sont requises pour la compréhension du chapitre 5.

### 1.4.1 Initialisation à partir de deux vues

#### 1.4.1.1 La géométrie épipolaire

La géométrie liée à l'observation d'une même scène rigide à travers deux points de vue différents est appelée *géométrie épipolaire*. Elle permet de définir la relation entre les projections  $\tilde{\mathbf{q}}$  sur la première image et  $\tilde{\mathbf{q}}'$  sur la deuxième, d'un point  $\tilde{\mathbf{Q}}$  de la scène observée. La connaissance de la structure de la scène n'est pas requise. Elle peut être calculée comme indiqué ci-dessous.

La *matrice fondamentale*  $F_o$  traduit de manière algébrique la géométrie épipolaire. A partir de cette matrice, on peut définir une contrainte très forte : la contrainte épipolaire. Elle existe pour tout couple de points  $\{\tilde{\mathbf{q}}, \tilde{\mathbf{q}}'\}$  résultant de la projection d'un même point 3D :

$$\tilde{\mathbf{q}}^T F_o \tilde{\mathbf{q}}' = 0.$$

Cette contrainte exprime le fait que chaque point doit se trouver sur la droite épipolaire donnée par son correspondant dans l'autre image. Les droites épipolaires se rencontrent sur les *épipôles* qui sont les noyaux de  $F_o$  :  $F_o \tilde{\mathbf{e}} = F_o^T \tilde{\mathbf{e}}' = \mathbf{0}$ .

La matrice fondamentale est liée à une reconstruction projective (non-calibrée) de la scène. Elle peut être estimée à partir d'un minimum de 7 correspondances de points, pour une estimation linéaire 8 correspondances sont requises (Hartley, 1997). Son équivalent pour une reconstruction Euclidienne, c'est-à-dire pour laquelle les paramètres internes de la caméra sont connus, est la matrice essentielle.

Notons que la géométrie épipolaire n'est pas valide lorsque la scène, observée par une seule caméra en mouvement, se déforme.

### 1.4.1.2 Estimation des caméras et de la structure

La formule générale pour une paire canonique de matrices de projection  $\{\tilde{\mathbf{P}}, \tilde{\mathbf{P}}'\}$  associée à une matrice fondamentale  $F_o$  est donnée par :

$$\tilde{\mathbf{P}} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \quad \text{et} \quad \tilde{\mathbf{P}}' = \begin{pmatrix} [\tilde{\mathbf{e}}']_{\times} F_o + \tilde{\mathbf{e}}' \mathbf{v}^T & \lambda \tilde{\mathbf{e}}' \end{pmatrix},$$

avec  $\mathbf{v}$  un vecteur quelconque de dimension 3,  $\lambda$  un scalaire non-nul et  $\tilde{\mathbf{e}}'$  l'épipôle associé à la deuxième image (il peut être calculé à partir de  $F_o$ ).

Une fois les matrices de projection connues, la structure est estimée par *triangulation* des points 3D. Ceci peut être fait, indépendamment pour chaque point  $\tilde{\mathbf{Q}}_j$ , par minimisation de l'erreur suivante :

$$\min_{\tilde{\mathbf{Q}}_j} d_A^2(\tilde{\mathbf{q}}, \tilde{\mathbf{P}}\tilde{\mathbf{Q}}_j) + d_A^2(\tilde{\mathbf{q}}', \tilde{\mathbf{P}}'\tilde{\mathbf{Q}}_j).$$

L'emploi de distances algébriques introduit une estimation linéaire des  $\tilde{\mathbf{Q}}_j$ , il est ensuite possible de raffiner l'estimation obtenue par minimisation non-linéaire d'une somme de distances Euclidiennes. Notons qu'un critère de distances Euclidiennes peut directement être minimisé (Hartley and Sturm, 1997). La *matrice essentielle* peut aussi être estimée uniquement à partir de correspondances de points. Les poses  $\{\mathbf{R}, \mathbf{t}\}$   $\{\mathbf{R}', \mathbf{t}'\}$  en sont alors extraites, voir par exemple (Huang and Faugeras, 1989).

### 1.4.1.3 Normalisation des données

Il a été montré dans (Hartley, 1997) que la normalisation des données améliore les résultats de la plupart des algorithmes linéaires comme par exemple l'estimation de la matrice fondamentale ou bien encore la triangulation de point. Le conditionnement numérique du système linéaire s'en trouve être amélioré. La normalisation consiste à translater le centre de gravité des points à l'origine du repère de référence et à appliquer un facteur d'échelle tel que la distance moyenne des points à l'origine soit  $\sqrt{2}$ . Par la suite, nous normaliserons les données dès lors que des distances algébriques sont utilisées.

## 1.4.2 Initialisation à partir de plusieurs vues

**L'approche séquentielle.** Son principe consiste à évaluer la caméra courante à partir de la structure connue à l'instant précédent. Une reconstruction initiale partielle est donc requise. Elle peut être obtenue à l'aide des deux premières vues. La structure observée est mise à jour pour chaque nouvelle caméra : des nouveaux points sont reconstruits et ceux déjà estimés sont éventuellement raffinés. Les étapes suivantes sont alors répétées :

- ▷ **Estimation linéaire** : la caméra  $\tilde{P}_i$  est évaluée à partir de  $p$  correspondances  $2D \leftrightarrow 3D$  entre la structure existante et les points détectés dans cette image (*resection*). L'erreur suivante est minimisée :

$$\min_{\tilde{P}_i} \sum_{j=1}^p d_A^2(\tilde{\mathbf{q}}_{i,j}, \tilde{P}_i \tilde{\mathbf{Q}}_j).$$

- ▷ **Triangulation de nouveaux points** : de nouveaux points observés sont reconstruits à condition qu'ils soient visibles dans la vue  $i$  et dans au moins une des vues précédentes.
- ▷ **Ajustement de faisceaux** (optionnel). L'estimation des caméras et des points 3D est raffinée par minimisation de distances Euclidiennes. Cette étape est décrite en §1.4.3.

Le principal inconvénient de cette approche est l'accumulation des erreurs au cours du temps, entraînant des dérives dans la reconstruction.

**L'approche hiérarchique.** Son principe est de diviser la vidéo en paires (ou triplets) d'images. La reconstruction est faite indépendamment sur chacune d'elles en utilisant les algorithmes de reconstruction pour 2 (ou 3) vues décrits précédemment. Les reconstructions partielles sont ensuite fusionnées. Elles doivent pour cela être exprimées dans le même référentiel. Une homographie  $\tilde{h}$  est alors estimée entre chaque reconstruction, par exemple par minimisation de l'erreur suivante :

$$\min_{\tilde{h}} \sum_{i=1}^{n'} \sum_{j=1}^m d_A^2(\tilde{\mathbf{q}}'_{i,j}, \tilde{P}'_i \tilde{h} \tilde{\mathbf{Q}}_j),$$

l'absence et la présence de " ' " indiquent respectivement le premier et le deuxième jeu d'images. L'utilisation de distances algébriques induit une solution linéaire pour l'estimation de  $\tilde{h}$ . Notons que ce critère n'est pas symétrique puisque les erreurs sont minimisées dans un jeu d'images mais pas dans l'autre. Une approche hiérarchique est décrit dans (Fitzgibbon and Zisserman, 1998).

**La factorisation.** Elle consiste à extraire de la *matrice de mesure*  $\mathcal{M}$ , rassemblant les points 2D, les matrices de mouvement  $W$  et de forme  $U$  telles que :  $\mathcal{M} = W_{kn \times r} U_{r \times m}$ , avec  $r \ll \min\{kn, m\}$ . Le choix du rang  $r$  dépend du modèle de projection utilisé :  $r = 3$  et  $r = 4$  respectivement pour un modèle de caméra affine et perspectif. Le principal inconvénient de la plupart des méthodes de *factorisation* est que tous les points doivent être apparents dans toutes les images :

- ▷ **Factorisation affine** : elle a été introduite dans (Tomasi and Kanade, 1992). Les coordonnées homogènes ne sont pas utilisées. La projection par caméra affine est donnée par :  $\mathbf{q}_{i,j} = \mathcal{A}_i \mathbf{Q}_j + \mathbf{t}_{A_i}$ . Au préalable, les points  $\{\mathbf{q}_{i,j}\}_{i,j=1}^{n,m}$  sont translatés sur leur centre de gravité (Tomasi and Kanade, 1992) :

$$\mathbf{q}_{i,j} \leftarrow \mathbf{q}_{i,j} - \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m \mathbf{q}_{i,j}$$

. Le système d'équations suivant est alors obtenu :

$$\mathcal{M}_{2n \times m} = \begin{pmatrix} \mathbf{q}_{1,1} & \mathbf{q}_{1,2} & \cdots & \mathbf{q}_{1,m} \\ \mathbf{q}_{2,1} & \mathbf{q}_{2,2} & \cdots & \mathbf{q}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_{n,1} & \mathbf{q}_{n,2} & \cdots & \mathbf{q}_{n,m} \end{pmatrix} = \begin{pmatrix} \mathcal{A}_1 \\ \mathcal{A}_2 \\ \vdots \\ \mathcal{A}_n \end{pmatrix}_{2n \times 3} \left( \mathbf{Q}_1 \quad \mathbf{Q}_2 \quad \cdots \quad \mathbf{Q}_m \right)_{3 \times m} + \Psi_{2n \times m},$$

avec  $\Psi$  une matrice modélisant le bruit sur les mesures. Une solution possible est donnée par la décomposition en valeur singulière  $\mathcal{M} = \Xi \Sigma \Gamma^T$  en ne conservant que les 3 vecteurs propres associés aux 3 plus grandes valeurs propres (Tomasi and Kanade, 1992) :  $J = \Xi_{2n \times 3} \Sigma_{3 \times 3}$  et  $Z = \Gamma_{m \times 3}^T$ . Cette solution n'est pas unique puisque n'importe quelle matrice inversible  $Q$  de taille  $3 \times 3$  peut être insérée dans la décomposition engendrant le même résultat :  $W = JQ$  et  $U = Q^{-1}Z \Rightarrow WU = (JQ)(Q^{-1}Z) = JZ$ .

- ▷ **Factorisation perspective** : elle est introduite dans (Sturm and Triggs, 1996). L'équation de projection perspective est définie à un facteur près. En écrivant explicitement ce facteur elle devient :  $w_{i,j} \tilde{\mathbf{q}}_{i,j} = \tilde{\mathbf{P}}_i \tilde{\mathbf{Q}}_j$ , avec la dernière composante de  $\tilde{\mathbf{q}}_{i,j}$  égale à 1. Le système suivant en découle :

$$\mathcal{M}_{3n \times m} = \begin{pmatrix} w_{1,1} \tilde{\mathbf{q}}_{1,1} & \cdots & w_{1,m} \tilde{\mathbf{q}}_{1,m} \\ w_{2,1} \tilde{\mathbf{q}}_{2,1} & \cdots & w_{2,m} \tilde{\mathbf{q}}_{2,m} \\ \vdots & \ddots & \vdots \\ w_{n,1} \tilde{\mathbf{q}}_{n,1} & \cdots & w_{n,m} \tilde{\mathbf{q}}_{n,m} \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \\ \vdots \\ \tilde{\mathbf{P}}_n \end{pmatrix}_{3n \times 4} \left( \tilde{\mathbf{Q}}_1 \quad \tilde{\mathbf{Q}}_2 \quad \cdots \quad \tilde{\mathbf{Q}}_m \right)_{4 \times m} + \Psi_{3n \times m}.$$

Sous l'hypothèse de coefficients  $\{w_{i,j}\}_{i,j=1}^{n,m}$  connus, les caméras ainsi que la structure sont estimées par une simple factorisation semblable à celle utilisée pour des caméras affines. Une solution est donnée par la décomposition en valeurs singulières de  $\mathcal{M}$  en ne conservant que les 4 vecteurs propres associés aux 4 plus grandes valeurs propres. La reconstruction obtenue n'est pas unique, elle est définie à une transformation perspective près. On parle de reconstruction perspective. Elle peut être étendue en une reconstruction Euclidienne si les caméras sont calibrées (Hartley and Zisserman, 2003).

Des solutions ont été proposées pour estimer les *profondeurs projective*  $\{w_{i,j}\}_{i,j=1}^{n,m}$  en utilisant la géométrie épipolaire (Martinec and Pajdla, 2005; Sturm and Triggs, 1996). Une alternative, proposée dans (Heyden, 1997), est de fixer tous les  $\{w_{i,j}\}_{i,j=1}^{n,m}$  à 1 puis de répéter les étapes suivantes jusqu'à convergence :

- \* Former la matrice  $\mathcal{M}$  étant donnés les  $\{w_{i,j}\}_{i,j=1}^{n,m}$ .
- \* Extraire les  $\{\tilde{\mathbf{P}}_i\}_{i=1}^n$  et  $\{\tilde{\mathbf{Q}}_j\}_{j=1}^m$  par SVD.
- \* Mettre à jour les  $\{w_{i,j}\}_{i,j=1}^{n,m}$  à partir des équations de projection  $w_{i,j} \tilde{\mathbf{q}}_{i,j} = \tilde{\mathbf{P}}_i \tilde{\mathbf{Q}}_j$ .

**Contraintes de fermeture.** Les *contraintes de fermeture* définissent des relations bilinéaires entre les matrices de projection et les matrices fondamentales. Elles sont proposées dans (Triggs, 1997) dans le cadre de caméras perspectives et dans (Guilbert et al., 2006) pour des caméras affines. L'accumulation de toutes ces contraintes permet d'estimer linéairement les matrices de projection. La structure 3D est ensuite estimée par triangulation. Notons que ces approches ne nécessitent pas que tous les points soient visibles dans toutes les vues mais uniquement qu'un nombre suffisamment important de matrices fondamentales puissent être estimées. Enfin, une contrainte de fermeture sur la structure est proposée dans (Tardif et al., 2007). Les caméras sont estimées par resection.

### 1.4.3 Ajustement de faisceaux

Soit  $\tilde{\mathbf{Q}}_j$  un point 3D de la scène observée exprimé en coordonnées homogènes,  $\tilde{\mathbf{P}}_i$  une matrice de projection de  $\mathbb{P}^3$  dans  $\mathbb{P}^2$  et  $\tilde{\mathbf{q}}_{i,j}$  le point 2D exprimé en coordonnées homogènes obtenu par :

$\tilde{\mathbf{q}}_{i,j} \sim \tilde{\mathbf{P}}_i \tilde{\mathbf{Q}}_j$ , avec  $i \in [1, \dots, n]$  les indices des images et  $j \in [1, \dots, m]$  les indices des points. De par la présence d'occultations certains points ne sont pas connus. Une carte de visibilité  $\mathcal{V}$  est définie :  $v_{i,j} = 1$  si le point 3D  $i$  est vu par la caméra  $j$ , sinon  $v_{i,j} = 0$ . Les caméras ainsi que les points 3D sont alors estimés en minimisant l'erreur de reprojection :

$$\min_{\{\tilde{\mathbf{P}}_i\}_{i=1}^n, \{\tilde{\mathbf{Q}}_j\}_{j=1}^m} \sum_{i=1}^n \sum_{j=1}^m v_{i,j} d^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{P}}_i \tilde{\mathbf{Q}}_j). \quad (1.2)$$

Ce critère peut être réécrit sous forme matricielle comme suit :

$$\min_{W, U} \|\mathcal{V} \odot (\mathcal{M} - WU)\|^2, \quad (1.3)$$

avec

$$\mathcal{M} = \begin{pmatrix} \tilde{\mathbf{q}}_{1,1} & \tilde{\mathbf{q}}_{1,2} & \cdots & \tilde{\mathbf{q}}_{1,m} \\ \tilde{\mathbf{q}}_{2,1} & \tilde{\mathbf{q}}_{2,2} & \cdots & \tilde{\mathbf{q}}_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\mathbf{q}}_{n,1} & \tilde{\mathbf{q}}_{n,2} & \cdots & \tilde{\mathbf{q}}_{n,m} \end{pmatrix}, \quad W = \begin{pmatrix} \tilde{\mathbf{P}}_1 \\ \tilde{\mathbf{P}}_2 \\ \vdots \\ \tilde{\mathbf{P}}_n \end{pmatrix} \quad \text{et} \quad U = \begin{pmatrix} \tilde{\mathbf{Q}}_1 & \tilde{\mathbf{Q}}_2 & \cdots & \tilde{\mathbf{Q}}_m \end{pmatrix}.$$

La matrice  $\mathcal{V}$  est définie par :

$$\mathcal{V} \stackrel{\text{def}}{=} \begin{pmatrix} v_{1,1} & \cdots & v_{1,m} \\ v_{1,1} & \cdots & v_{1,m} \\ \vdots & \cdots & \vdots \\ v_{n,1} & \cdots & v_{n,m} \\ v_{n,1} & \cdots & v_{n,m} \end{pmatrix}.$$

La minimisation de l'équation (1.2) est appelée *ajustement de faisceaux*. Les équations sont non-linéaires par rapport aux paramètres. Leur estimation se fait habituellement par l'algorithme Levenberg-Marquardt, décrit en §1.6.2. La convergence vers le minimum global n'étant pas garantie, une initialisation proche du minimum global est fortement recommandée même si une initialisation aléatoire des paramètres peut donner de bons résultats (Buchanan and Fitzgibbon, 2005). Plusieurs points de départ doivent alors être testés, la meilleure solution est finalement conservée. Ce procédé s'avère être coûteux en terme de temps de calcul. L'ajustement de faisceaux constitue souvent la dernière étape d'un module de reconstruction 3D. Notons que si les paramètres internes des caméras sont connus, on parle d'ajustement de faisceaux Euclidien. Dans le cas contraire, on parle d'ajustement de faisceaux projectif.

Une autre possibilité est d'alterner l'estimation de la *matrice de mouvement*  $W$  et de la *matrice de forme*  $U$  (c'est-à-dire d'alterner l'estimation des caméras et des points 3D) en supposant que l'une des deux matrices est connue (Hartley and Schaffalitzky, 2003). Cette approche appelée *alternation*, utilise le fait que si l'une des matrices est connue alors l'estimation de la deuxième est plus facile.

## 1.5 Images, transformations et augmentation

### 1.5.1 Transformation entre images

Nous considérons deux images, l'image de référence et celle à recalculer (autrement appelée image cible ou image courante). Elles sont respectivement notées  $\mathcal{I}_0$  et  $\mathcal{I}_i$ . La transformation géométrique

définie entre ces images est notée  $\mathcal{W}$ , elle est appelée par la suite *fonction de déformation*. Elle dépend d'un vecteur de paramètres  $\mathbf{u}$  et transfère un pixel  $\mathbf{q} \in \mathcal{R}$  de l'image de référence vers l'image cible :

$$\mathbf{q}_i = \mathcal{W}(\mathbf{q}; \mathbf{u}),$$

avec  $\mathcal{R}$  l'ensemble des pixels d'intérêt.  $\mathbf{q}_i$  ne correspond pas forcément à un pixel de l'image cible. Afin de lui affecter une intensité  $\mathcal{I}_i(\mathbf{q}_i)$ , il est nécessaire d'interpoler l'image  $\mathcal{I}_i$ . Différents processus d'interpolation peuvent être utilisés comme par exemple l'interpolation bilinéaire ou bicubic. Cette dernière est utilisée lors des différentes expérimentations. Elle est plus lisse et introduit moins d'artefacts que l'interpolation bilinéaire. L'intensité associée à un pixel est exprimé en Unité d'Intensité (UI), elle est comprise entre 0 et 255. Le recalage d'images est étudié aux chapitres 2, 3, 4. Notons que l'indice  $i$  relatif aux images courantes est la plupart du temps omis dans la suite du tapuscrit pour des raisons de clarté.

Nous notons  $\frac{\partial \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q}; \mathbf{u})$  et  $\frac{\partial_a \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q}; \mathbf{u})$ , les dérivées partielles et les dérivées directionnelles partielles suivant la direction  $\mathbf{d} \in \mathbb{S}^1$  de la transformation  $\mathcal{W}$ , avec  $\mathbb{S}^1$  le cercle de rayon unité. Leur approximation par différences finies sont respectivement notées  $\mathbf{E}(\mathbf{q}; \mathbf{u})$  et  $\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})$ . Par exemple  $\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u}) = \frac{\mathcal{W}(\mathbf{q}+\epsilon \mathbf{d}; \mathbf{u}) - \mathcal{W}(\mathbf{q}-\epsilon \mathbf{d}; \mathbf{u})}{2\epsilon}$ . Dans le tapuscrit, les dérivées centrale, à gauche et à droite sont distinguées par l'ajout respectif des indices  $c, l, r$ . L'absence d'indice signifie qu'une dérivée centrale est utilisée. Enfin, le gradient d'une image, c'est-à-dire sa dérivée partielle par rapport à  $\mathbf{q}$ , est noté  $\nabla \mathcal{I}$  et défini par :  $\nabla \mathcal{I}(\mathbf{q}) \stackrel{\text{def}}{=} \frac{\partial \mathcal{I}}{\partial \mathbf{q}}(\mathbf{q})$ .

**Exemples de transformations rigides.** Le choix de la transformation géométrique dépend de l'application visée. Nous nous intéressons plus particulièrement aux transformations non-rigides. Celles que nous utilisons dans notre étude sont décrites en §2.3. Nous présentons ici des exemples de transformations rigides.

▷ **Similitudes** : Elles sont décrites par :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \begin{pmatrix} u_1 \cos(u_2) q^x - \sin(u_2) q^y + u_3 \\ u_1 \sin(u_2) q^x + \cos(u_2) q^y + u_4 \end{pmatrix}.$$

Elles conservent les rapports de longueurs, les angles et plus particulièrement le parallélisme.

▷ **Affinités** : Elles sont décrites par :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \begin{pmatrix} (1 + u_1) q^x + u_3 q^y + u_5 \\ u_2 q^x + (1 + u_4) q^y + u_6 \end{pmatrix}.$$

Elles conservent notamment le parallélisme.

▷ **Homographies** : Elles sont données par :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \frac{1}{1 + u_7 q^x + u_8 q^y} \begin{pmatrix} (1 + u_1) q^x + u_3 q^y + u_5 \\ u_2 q^x + (1 + u_4) q^y + u_6 \end{pmatrix}.$$

Elles décrivent des transformations linéaires pour des points exprimés en coordonnées homogènes. Elles conservent le *bi-rapport*.

## 1.5.2 Augmentation 2D d'une image

L'*augmentation 2D* ("retexturing" en anglais) consiste à superposer un logo<sup>4</sup> à une image. Des exemples d'augmentation 2D d'images d'une surface déformable sont présentés aux chapitres 3 et 4. Pour que le rendu soit visuellement réaliste, il faut que les transformations géométriques entre les images soient estimées précisément. Cette problématique est étudiée dans les chapitres 2, 3 et 4. La méthodologie utilisée pour l'augmentation 2D d'une image est décrite ci-dessous. La transformation entre l'image de référence et l'image courante, ainsi que son inverse, sont supposées connues. Nous disposons également d'une image  $\mathcal{L}_0$  de taille semblable à celle de l'image de texture, sur laquelle se trouve le logo qui va servir pour l'augmentation. Elle est transformée par  $\mathcal{W}^{-1}$  en l'image  $\mathcal{L}$  :

$$\mathcal{L}(\mathbf{q}) \leftarrow \mathcal{L}_0(\mathcal{W}(\mathbf{q}; \mathbf{u})^{-1}).$$

L'augmentation 2D consiste alors en une addition pondérée de deux images. La pondération se fait par une carte d'occupation  $\mathcal{H}^o$  dont les éléments sont compris entre 0 ( $\Leftrightarrow$  logo absent) et 1 ( $\Leftrightarrow$  logo présent). Des valeurs non binaires sont affectées aux bords du logo, voir ci-dessous.

L'image augmentée est donnée par :

$$\mathcal{I} \leftarrow \mathcal{I} \odot (1 - \sigma_t \mathcal{H}^o) + \mathcal{L} \odot \sigma_t \mathcal{H}^o,$$

où  $\sigma_t$  contrôle la transparence du logo. La figure 1.2 illustre le principe de l'augmentation 2D d'une image.

La construction de la carte d'occupation  $\mathcal{H}^o$  se fait dans un premier temps par segmentation « grossière » de l'image  $\mathcal{L}$ , séparant ainsi le logo de son arrière plan. Des algorithmes évolués de segmentation peuvent être utilisés. Cependant, un simple seuillage des trois canaux RVB<sup>5</sup> est suffisant sous l'hypothèse d'un arrière plan uniforme dont la couleur n'est pas présente dans le logo. Les pixels segmentés comme faisant partie du logo sont mis à 1 dans  $\mathcal{H}^o$  et les autres à 0. Ensuite, la carte d'occupation est raffinée par une opération d'érosion afin de palier aux erreurs de segmentation présentes sur les bords. Une transformée en distance est appliquée à la carte  $\overline{\mathcal{H}}^o = 1 - \mathcal{H}^o : \overline{\mathcal{H}}_D^o$ . Les pixels  $\mathbf{q}^b$  correspondant aux bords du logo sont détectés par simple seuillage (seuil  $\sigma_d$ ) de  $\overline{\mathcal{H}}_D^o$ .

La carte d'occupation peut être mise à jour par  $\mathcal{H}^o(\mathbf{q}^b) = 0$ . Cette opération introduit des transitions abruptes qui sont gênantes visuellement. Afin d'améliorer le rendu, les transitions sont lissées. La carte d'occupation est donnée par  $\mathcal{H}^o \leftarrow \exp(-(\overline{\mathcal{H}}_D^o - \sigma_d)^2 / \sigma_g^2)$  avec  $\sigma_d > \sigma_g$ . Au préalable, on affecte la valeur  $\sigma_d$  dans  $\overline{\mathcal{H}}_D^o$  aux pixels dont la distance est supérieure à  $\sigma_d$ . La construction de la carte d'occupation est illustrée sur la figure 1.3.

## 1.5.3 Augmentation 3D d'une image

Nous appelons *augmentation 3D* l'ajout d'objets 3D virtuels dans les images. Cette opération requiert une connaissance précise de la structure de la scène observée ainsi que du déplacement des caméras. Pour un rendu réaliste, il est nécessaire de détecter les régions de l'objet 3D qui ne sont pas observées par la caméra courante. Pour cela des algorithmes comme le z-buffer peuvent être utilisés. Des techniques de rendu 3D sont également appliquées, elles retranscrivent les variations d'illumination : ombres, spéularités, *etc.*, en fonction de la position de la caméra par rapport à une ou plusieurs sources lumineuses. Des exemples d'augmentation 3D d'images d'une surface déformable sont présentées en §5.5.2.

<sup>4</sup>Une vidéo, du texte, *etc.* peuvent également être superposés.

<sup>5</sup>Rouge, Vert et Bleu.



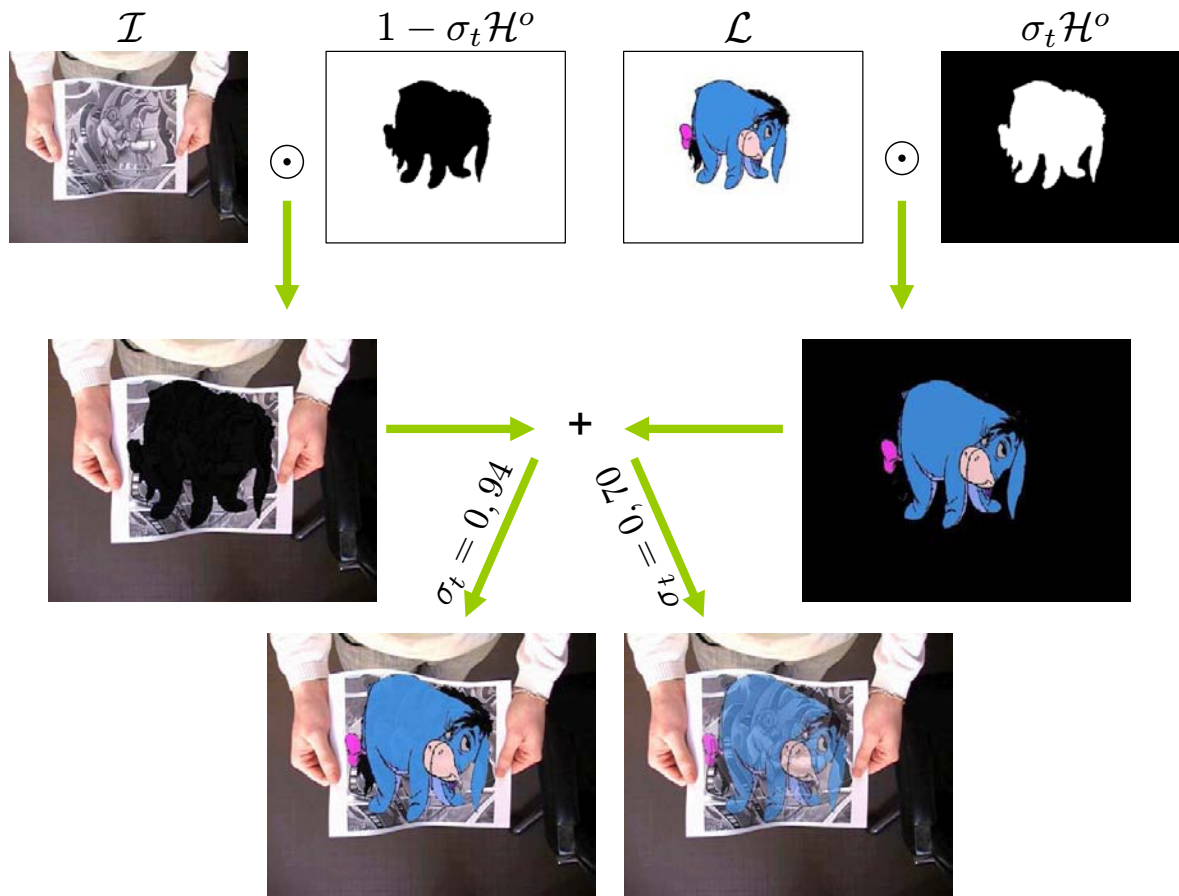


FIG. 1.2 – Principe de l'augmentation 2D d'une image.

## 1.6 Optimisation numérique

Dans notre étude, les fonctions de coût à minimiser sont (ou peuvent être mises) sous la forme d'une somme d'éléments au carré :

$$\mathcal{E}(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{i=1}^p f_i^2(\mathbf{u}),$$

avec  $f_i(\mathbf{u})$  le  $i$ ème résidu,  $p$  le nombre de résidus et  $\mathbf{u}$  le vecteur de paramètres. En concaténant l'ensemble des résidus  $\mathbf{f}^T(\mathbf{u}) = (f_1(\mathbf{u}) \ \cdots \ f_m(\mathbf{u}))$ , on obtient  $\mathcal{E}(\mathbf{u}) = \|\mathbf{f}(\mathbf{u})\|^2$ .

On cherche à estimer le vecteur de paramètres  $\hat{\mathbf{u}}$  tel que  $\mathcal{E}(\hat{\mathbf{u}}) \rightarrow \min$ . En pratique, les erreurs liées aux approximations numériques, au bruit présent dans les images et à la modélisation approximative de la physique, font que le critère ne s'annule pas. Le vecteur de paramètres  $\hat{\mathbf{u}}$  pour lequel l'erreur  $\mathcal{E}(\hat{\mathbf{u}})$  est minimale est recherché, c'est-à-dire :

$$\hat{\mathbf{u}} = \arg \min_{\mathbf{u}} \mathcal{E}(\mathbf{u}).$$

Les résidus  $f_i(\mathbf{u})$  peuvent être des fonctions linéaires ou non-linéaires de  $\mathbf{u}$ , introduisant respectivement un problème d'optimisation de moindres carrés linéaires ou non-linéaires. Dans le premier cas la solution est obtenue par SVD. Dans le deuxième cas l'estimation des paramètres se fait itérativement en faisant l'hypothèse que les  $f_i$  sont localement linéaires par rapport au vecteur de paramètres. A

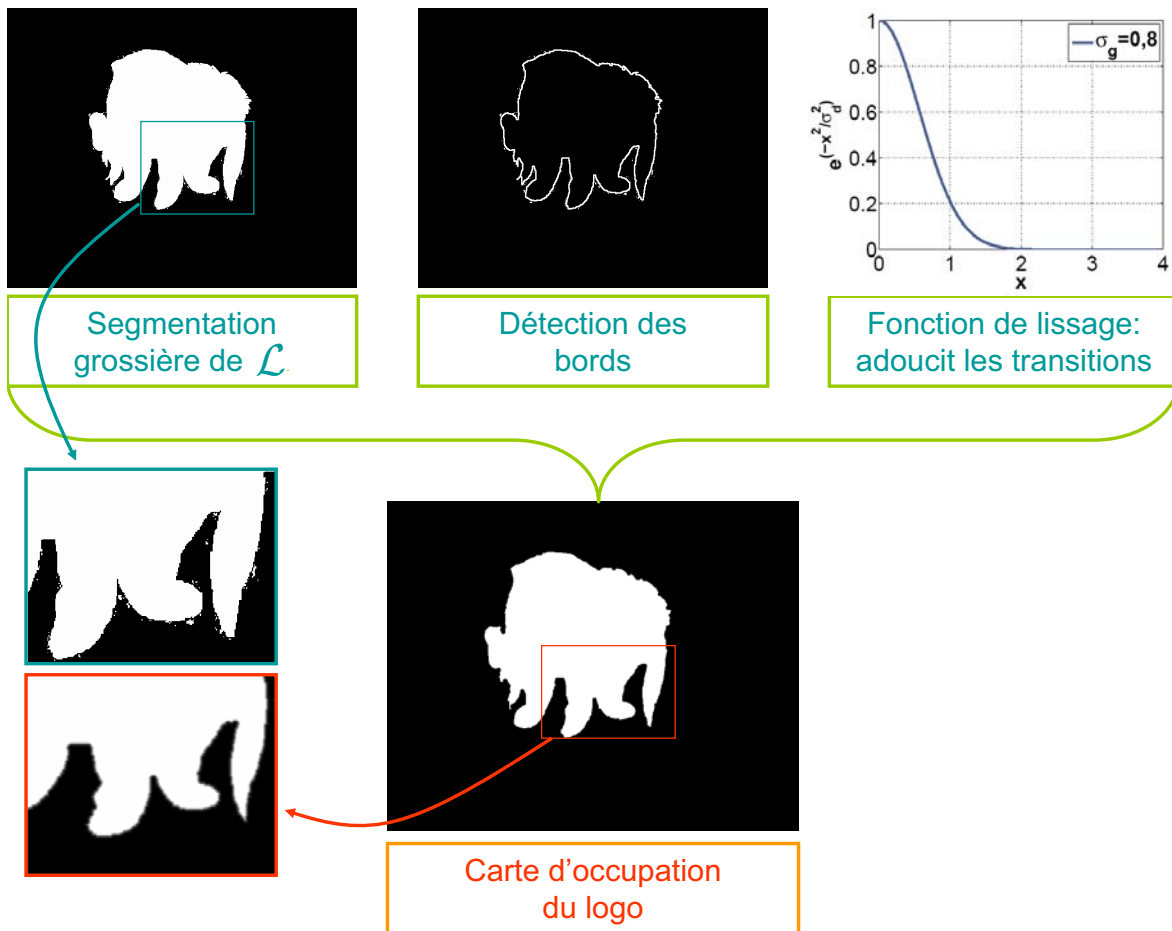


FIG. 1.3 – **Construction de la carte d’occupation.** Les bords du logo sont détectés par seuillage de la carte de distance. Celle-ci est issue d’une transformée en distance appliquée à la segmentation grossière de  $\mathcal{L}$ . Une fonction de lissage est ensuite appliquée afin d’éliminer les imperfections de segmentation et d’améliorer le rendu de l’augmentation 2D.

chaque itération un incrément  $\Delta \mathbf{u}$  est estimé. Ce dernier permet de mettre à jour le vecteur de paramètres courant tout en vérifiant la décroissance de l’erreur. A partir d’une estimation initiale, on obtient ainsi une suite de valeurs qui converge vers un minimum local  $\hat{\mathbf{u}}$  de  $\mathcal{E}$ . Le critère de convergence est fixé par  $\|\Delta \mathbf{u}\| \leq \epsilon$ , où  $\epsilon$  est typiquement fixé à  $\epsilon = 10^{-8}$

### 1.6.1 Moindres carrés linéaires

Si les  $f_i$  sont des fonctions linéaires de  $\mathbf{u}$ , le critère peut être réécrit sous forme matricielle. Dans le cas homogène, c’est-à-dire sans membre droit, nous obtenons :

$$\mathcal{E}(\mathbf{u}) = \|\mathbf{F}\mathbf{u}\|^2.$$

La solution au sens des moindres carrés, sous la contrainte que  $\|\mathbf{u}\|^2 = 1$ , est donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice  $\mathbf{F}$  obtenue par SVD.

Certains problèmes examinés dans ce tapuscrit conduisent à la résolution d’un système non-homogène dont la solution au sens des moindres carrés est donnée par la minimisation du critère :

$\mathcal{E}(\mathbf{u}) = \|\mathbf{F}\mathbf{u} - \mathbf{b}\|^2$ , avec  $\mathbf{b} \neq \mathbf{0}$ . Notons que le nombre d'équations est dans tous les cas supérieur au nombre d'inconnues. La résolution du système d'équations normales  $\mathbf{F}^T\mathbf{F}\mathbf{u} = \mathbf{F}^T\mathbf{b}$  conduit à la pseudo inverse :

$$\mathbf{u} = \mathbf{F}^\dagger\mathbf{b}.$$

## 1.6.2 Moindres carrés non-linéaires

Les différentes méthodes d'optimisation décrites ci-dessous diffèrent du point de vue du calcul de l'incrément  $\Delta\mathbf{u}$ . Elles sont habituellement couplées à une mise à jour additive  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$  du vecteur de paramètres. Notons que pour certaines fonctions de coût une mise à jour compositionnelle dans l'espace des transformations peut être envisagée, plus de détails sont donnés en §2.2.2.3.

**Descente de gradient.** Cette méthode fixe la direction de descente comme la ligne de plus grande pente de  $\mathcal{E}$ . Elle est basée sur l'observation que  $\mathcal{E}(\mathbf{u})$  décroît le plus rapidement dans la direction opposée à celle du gradient de  $\mathcal{E}$ . Dans le cadre d'une fonction à plusieurs paramètres, les dérivées partielles sont regroupées dans un vecteur : le gradient de la fonction  $\mathbf{g}_i(\mathbf{u})$ . Chaque élément correspond alors à la dérivée partielle de la fonction selon l'un de ses paramètres. En rassemblant l'ensemble des vecteurs gradients associés à chaque résidus  $f_i$ , on obtient la matrice *Jacobienne* de l'erreur :

$$\mathcal{J} = \begin{pmatrix} \mathbf{g}_1^T \\ \vdots \\ \mathbf{g}_m^T \end{pmatrix}.$$

L'incrément de la descente de gradient est donné par :  $\Delta\mathbf{u} = -\mathcal{J}^T\mathbf{f}$ . L'avantage de cette méthode est que la convergence vers un minimum local est assurée. En revanche elle converge habituellement en un très grand nombre d'itérations, notamment aux abords de minima locaux. Ceci s'explique en partie par la non prise en compte de l'information de courbure liée aux dérivées secondes.

**Gauss-Newton.** L'approximation locale par Gauss-Newton (GN) de  $\mathcal{E}(\mathbf{u})$  est obtenue par développement de Taylor au premier ordre de  $\mathbf{f}(\mathbf{u} + \Delta\mathbf{u})$  :

$$\begin{aligned} \mathcal{E}(\mathbf{u} + \Delta\mathbf{u}) &= \|\mathbf{f}(\mathbf{u} + \Delta\mathbf{u})\|^2 \\ &= \|\mathbf{f}(\mathbf{u}) + \mathcal{J}\Delta\mathbf{u}\|^2, \end{aligned}$$

où  $\mathcal{J}$  est la matrice Jacobienne de  $\mathbf{f}$ . Une itération de Gauss-Newton revient à résoudre un système de moindres carrés linéaires non-homogènes. L'incrément est obtenu par résolution des équations normales :

$$\mathbf{H}\Delta\mathbf{u} = -\mathbf{b}, \tag{1.4}$$

avec  $\mathbf{H} = \mathcal{J}^T\mathcal{J}$  l'approximation de Gauss-Newton de la matrice *Hessienne* et  $\mathbf{b} = \mathcal{J}^T\mathbf{f}$ . Au final l'incrément de GN est donné par :

$$\Delta\mathbf{u} = -\mathbf{H}^{-1}\mathcal{J}^T\mathbf{f}.$$

L'approximation de Gauss-Newton assure une convergence quadratique efficace, sans estimer la « vrai » matrice Hessienne. La décroissance du critère n'est cependant pas assurée. Le comportement de la convergence dépend fortement de la condition initiale.

**Levenberg-Marquardt.** L'algorithme Levenberg-Marquardt (LM) fut publié dans (Levenberg, 1944) puis redécouvert dans (Marquardt, 1963). Il est devenu un standard parmi les méthodes d'optimisation de moindres carrés non-linéaires. Chaque itération combine la méthode de GN et la méthode de descente de gradient (décrites ci-dessus). Lorsque la solution est éloignée d'un minimum local, la descente de gradient est privilégiée, offrant une convergence lente mais assurée. Au contraire, aux abords d'un minimum local, la méthode de GN est utilisée de par sa plus grande efficacité de convergence. L'incrément est donné par  $\Delta \mathbf{u} = -(\mathcal{J}^T \mathcal{J} + \tau \mathbf{I})^{-1} \mathcal{J}^T \mathbf{f}$ , avec  $\mathcal{J}$  la matrice Jacobienne de  $\mathbf{f}$ .

Par rapport à l'incrément de GN, seule la diagonale de la matrice  $\mathbf{H} = \mathcal{J}^T \mathcal{J}$  est modifiée. Cette correction est appelée *amortissement* et le paramètre  $\tau$  est dit *coefficient d'amortissement*. Augmenter ou réduire  $\tau$  revient respectivement à donner plus ou moins d'importance à la descente de gradient.

- ▷ Si l'incrément  $\Delta \mathbf{u}$  introduit une diminution de l'erreur, alors il est accepté et  $\tau$  est divisé par 10 avant la prochaine itération.
- ▷ Au contraire, si  $\Delta \mathbf{u}$  entraîne une augmentation de l'erreur, alors  $\tau$  est multiplié par 10 et l'incrément est recalculé. Ces opérations sont répétées jusqu'à ce que l'erreur diminue.

Si le coefficient  $\tau$  est grand, alors la matrice  $(\mathcal{J}^T \mathcal{J} + \tau \mathbf{I})$  est proche d'une matrice diagonale et  $\Delta \mathbf{u}$  correspond quasiment à l'incrément d'une descente de gradient. Par contre, si  $\tau$  est petit, il se rapproche de celui de Gauss-Newton. L'algorithme est adaptatif et contrôle automatiquement l'amortissement. De ce fait, il est capable d'alterner entre une lente descente de gradient loin d'un minimum local et une rapide convergence quadratique dans son voisinage.

**"Efficient Second order Minimisation".** L'approximation Efficient Second order Minimisation (ESM) est proposé dans (Malis, 2004). Elle s'avère être spécifique aux fonctions de coût pour lesquelles les résidus  $f_i$  peuvent être mis sous la forme suivante :  $f_i(\mathbf{u}) = s_i(\mathbf{u}_0) - s_i(\mathbf{u})$ , avec  $\mathbf{u}_0$  les paramètres identité (ou de référence) connus et fixes. Les fonctions de coût utilisées dans notre étude sont (ou peuvent être mises), la plupart du temps sous cette forme. L'erreur finale s'écrit :

$$\mathcal{E}(\mathbf{u}) = \sum_{i=1}^p \|s_i(\mathbf{u}_0) - s_i(\mathbf{u})\|^2.$$

L'approximation de la fonction de coût par ESM est une approximation du second ordre, théoriquement meilleure que celle de GN, ne nécessitant pas le calcul de la « vrai » matrice Hessienne. Elle est donnée par :

$$\mathcal{E}(\mathbf{u} + \Delta \mathbf{u}) = \sum_{i=1}^p \left\| s_i(\mathbf{u}_0) - s_i(\mathbf{u}) + \frac{1}{2}(\mathbf{g}_i^T(\mathbf{u}) + \mathbf{g}_i^T(\mathbf{u}_0))\Delta \mathbf{u} \right\|^2,$$

où les  $\mathbf{g}_i(\mathbf{u})$  sont les vecteurs gradients des  $s_i(\mathbf{u})$  et  $\mathbf{g}_i(\mathbf{u}_0)$  les vecteurs gradients des  $s_i(\mathbf{u}_0)$ . L'incrément de ESM est donné par  $\Delta \mathbf{u} \stackrel{\text{def}}{=} -2(\mathcal{J}(\mathbf{u}_0) + \mathcal{J}(\mathbf{u}))^\dagger \mathbf{f}$ .

Cette procédure d'optimisation possède des propriétés de convergence intéressantes. Par exemple le nombre d'itérations pour converger est souvent inférieur à celui de l'algorithme Gauss-Newton.

# Chapitre 2

## Cadre général du recalage d'images par modèles déformables

---

---

Nous définissons les éléments de base pour le recalage d'images d'une surface déformable. La première partie de ce chapitre est une revue des méthodes classiques estimant les modèles déformables à partir de données image. Les deuxième et troisième parties décrivent les modèles déformables utilisés au cours de nos travaux.

---

---

### 2.1 Introduction

Le principe du recalage d'images est d'estimer les transformations qui rendent deux ou plusieurs images les plus semblables possibles. La ressemblance est mesurée par des critères que nous présentons en §2.2. Ces transformations sont principalement de deux types :

- ▷ **Une transformation géométrique modifiant la position des pixels dans les images.** Elle engendre un champ de déplacement dense entre une image de référence et l'image cible. Nous nous intéressons plus particulièrement aux transformations non-rigides, appelées par la suite fonctions de déformation. Celles utilisées au cours de nos travaux sont décrites en §2.3.
- ▷ **Une transformation photométrique modifiant la valeur des pixels.** Elle modélise les variations d'illumination globales et / ou locales. Notons que ces phénomènes ne sont pas toujours explicitement modélisés : certaines approches privilégient une estimation de la transformation géométrique à partir d'informations image invariantes aux conditions d'éclairage.

Dans la littérature, il existe deux grandes approches de recalage d'images. L'approche *directe* utilise l'information de couleur contenue dans chaque pixel des images. Elle minimise une erreur basée sur l'information directement disponible, comme par exemple l'intensité lumineuse. Par opposition aux approches *basées primitives*<sup>1</sup> qui extraient au préalable un ensemble discret de primitives (points, droites, courbes, *etc.*) indépendamment dans chaque image, puis analysent les correspondances afin d'estimer le déplacement. Ces méthodes minimisent une fonction de coût basée sur la notion de distance entre primitives correspondantes.

---

<sup>1</sup>"feature-based" en anglais.

*Propriétés des approches directes.* Elles estiment précisément les transformations entre images de par la grande quantité d'information disponible. De plus, elles ne requièrent pas la mise en correspondance de primitives. En contrepartie, elles sont sensibles aux variations d'apparences et ne permettent pas l'estimation de mouvements importants entre des images puisque l'information d'intensité n'est que localement corrélée aux déplacements.

*Propriétés des approches basées primitives.* Elles ont pour principaux avantages d'estimer des déplacements importants entre les images. L'évaluation des transformations géométriques est en outre peu sensible aux variations d'éclairément. En contrepartie, elles nécessitent de détecter et de mettre en correspondance des primitives. Le nombre de correspondances obtenu peut être faible, il en résulte alors une estimation des transformations peu précise.

Ces deux types de méthodes s'avèrent être complémentaires. Leur combinaison permet de bénéficier des avantages des deux approches (Ladikos et al., 2007; Pilet et al., 2007). Dans un premier temps, le déplacement est estimé à partir de primitives puis est raffiné par minimisation d'un critère direct. L'algorithme introduit gère alors les grands déplacements tout en estimant précisément la transformation géométrique pour chaque pixel. Dans notre étude, nous utilisons principalement les méthodes directes. Les approches basées primitives ne sont que brièvement introduites en §2.2.3. Plus de détails sur ces méthodes peuvent être trouvés dans la littérature (Lepetit et al., 2005; Pilet et al., 2005; Torr and Zisserman, 1999).

Le recalage d'images d'une surface déformable à partir de méthodes directes et basées primitives est intrinsèquement mal posé. Il est nécessaire de contraindre le problème par l'emploi de modèles déformables et / ou en incorporant un ou plusieurs terme(s) de régularisation dans la fonction de coût. Une grande variété de modèles déformables sont proposés dans la littérature. Le choix se fait en fonction de l'application visée et des propriétés de la surface observée. Par exemple, certains modèles engendrent des déformations lisses, d'autres introduisent des déformations ponctuelles.

**Organisation du chapitre.** Ce chapitre s'articule autour de trois grandes parties. En §2.2 les outils permettant d'estimer une fonction de déformation à partir de données images sont présentés : les différents critères, les termes de régularisation, les méthodes spécifiques d'optimisation *etc.* La deuxième partie de ce chapitre, §2.3, est consacrée à l'étude des fonctions de déformation utilisées au cours du tapuscrit. Nous présentons ensuite en §2.4 les modèles approximant les déformations 3D d'une surface par une combinaison linéaire de modes de déformation. Enfin, nous concluons ce chapitre en §2.5.

## 2.2 Cadre général de l'estimation

### 2.2.1 Forme de la fonction de coût

En recalage non-rigide, la forme générale de la fonction de coût se compose de deux termes, un *terme de données*  $\mathcal{E}_d$  qui compare les informations issues de l'image de référence et celles provenant de l'image cible, et un *terme de régularisation*  $\mathcal{E}_s$  :

$$\mathcal{E}(\mathbf{u}) \stackrel{\text{def}}{=} \mathcal{E}_d(\mathbf{u}) + \lambda_s \mathcal{E}_s(\mathbf{u}), \quad (2.1)$$

avec  $\lambda_s$  le paramètre contrôlant l'influence de la régularisation. Le terme  $\mathcal{E}_s$  est important puisque l'estimation du champ de déplacement dense entre deux images à partir uniquement d'intensités lumineuses ou de primitives est un problème mal posé (Horn and Schunck, 1981) si la fonction de déformation utilisée n'est pas suffisamment contrainte. Notons que certaines fonctions de déformation sont naturellement régularisées : par exemple les *fonctions plaque mince* (TPS<sup>2</sup>) (Bookstein, 1989) minimisent l'énergie de courbure. Dans ce cas le terme de régularisation  $\mathcal{E}_s$  peut être omis. Il est également possible d'ajouter d'autres termes à la fonction de coût afin d'imposer certaines propriétés aux déformations estimées, voir par exemple en §4.4.

### 2.2.1.1 Critères directs

**Critère de type somme de différences au carré.** Le terme de données le plus répandu pour les approches directes est un critère de type *somme de différences au carré* (SSD<sup>3</sup>) calculé entre l'image de référence et l'image courante recalée sur l'image de référence par la fonction de déformation  $\mathcal{W}(\mathbf{q}; \mathbf{u})$  :

$$\mathcal{E}_d(u) \stackrel{\text{def}}{=} \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) - \mathcal{I}_0(\mathbf{q})\|^2. \quad (2.2)$$

La figure 2.1(b) illustre schématiquement ce critère. La variation d'intensité est supposée être introduite uniquement par le déplacement entre les deux images : c'est l'hypothèse de constance d'intensité<sup>4</sup>. Les conditions d'éclairage sont présumées constantes au cours du temps. Afin de relaxer cette hypothèse, une transformation photométrique peut être explicitement incorporée dans l'équation (2.2) comme décrit en §2.2.2.1. Dans la suite du tapuscrit, notamment aux chapitres 3 et 4, nous utiliserons ce critère pour le recalage d'images d'une surface déformable.

**L'Information Mutuelle.** L'*information mutuelle* (MI<sup>5</sup>) est la quantité d'information d'une image contenue dans une deuxième image. La MI est maximale lorsque les deux images sont identiques. Il en existe plusieurs définitions, l'une d'entre elle est donnée par :

$$\mathcal{E}_d(\mathbf{u}) = \zeta(\mathcal{I}_0) + \zeta(\mathcal{I}(\mathcal{W}(\cdot; \mathbf{u}))) - \zeta(\mathcal{I}_0, \mathcal{I}(\mathcal{W}(\cdot; \mathbf{u}))), \quad (2.3)$$

avec  $\zeta$  l'*entropie de Shannon* (Shannon, 1948). Elle mesure la quantité d'information contenue dans une série d'événements. L'entropie d'une image traduit sa complexité. Une image uniforme a une entropie faible tandis qu'une image fortement texturée possède une entropie importante. L'entropie jointe  $\zeta(\mathcal{I}_0, \mathcal{I}(\mathcal{W}(\cdot; \mathbf{u})))$  mesure la quantité d'information que les images  $\mathcal{I}_0$  et  $\mathcal{I}(\mathcal{W}(\cdot; \mathbf{u}))$  ont en commun. Si elles sont similaires, l'entropie jointe est minimale. Elle est calculée à partir de l'histogramme joint des deux images. Maximiser la MI revient donc à dire que le maximum d'information est recherché dans une image, tout en cherchant à les faire se ressembler le plus possible. Contrairement aux critères précédents, la MI ne se base pas directement sur une différence d'intensité. Elle est bien adaptée pour le recalage d'images issues de différents capteurs (multimodale), ce qui explique son succès en imagerie médicale (Pluim et al., 2003; Rueckert et al., 1999).

<sup>2</sup>"Thin Plate Spline" en anglais

<sup>3</sup>"Sum of Square Difference" en anglais

<sup>4</sup>"Brighness Constancy" en anglais

<sup>5</sup>"Mutual Information" en anglais

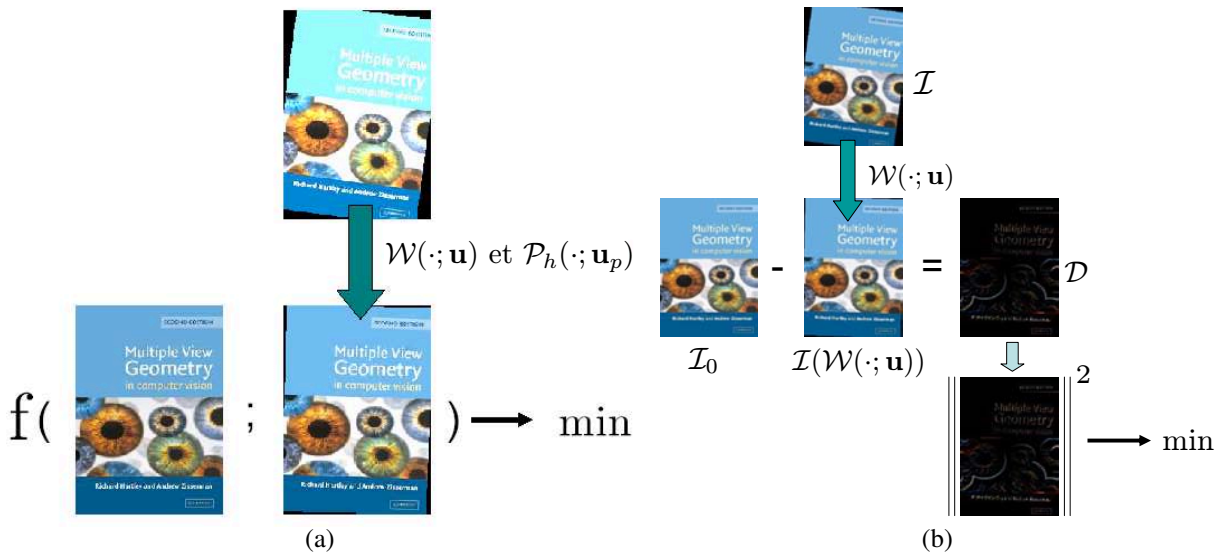


FIG. 2.1 – **Principe des méthodes directes de recalage d'images.** (a) Elles minimisent un critère dépendant de l'information de couleur contenue par chaque pixel des images. (b) Le critère de type SSD minimise la norme de l'image de différence  $\mathcal{D}$ , obtenue en soustrayant l'image de référence  $\mathcal{I}_0$  et l'image cible  $\mathcal{I}$  recalée sur cette dernière par la transformation géométrique  $\mathcal{W}(\cdot; \mathbf{u})$ .

### 2.2.1.2 Critères basés primitives

Le critère de base des approches basées primitives minimise la distance Euclidienne entre les primitives extraites de l'image de référence et de l'image cible. Elles sont au préalable mises en correspondance :

$$\mathcal{E}_d(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{\mathbf{q} \leftrightarrow \mathbf{q}' \in \Theta} w_c d^2(\mathbf{q}', \mathcal{W}(\mathbf{q}; \mathbf{u})), \quad (2.4)$$

où  $\Theta$  est l'ensemble des primitives mises en correspondance et  $w_c \in [0, 1]$  est un poids associé à chaque correspondance.

### 2.2.1.3 Termes de régularisation

Une grande variété de termes de régularisation est proposée dans la littérature. Une possibilité est d'imposer des déformations lisses. On parle alors de *terme de lissage*. Ils sont construits à partir des dérivées de la fonction de déformation, comme par exemple le terme de lissage quadratique d'ordre 1 proposé dans (Horn and Schunck, 1981) :

$$\mathcal{E}_s \stackrel{\text{def}}{=} \int_{\mathcal{R}} \int_{\mathcal{R}} \left( \frac{\partial \mathcal{W}}{\partial x} \right)^2 + \left( \frac{\partial \mathcal{W}}{\partial y} \right)^2 dx dy, \quad (2.5)$$

ou le terme de lissage quadratique d'ordre 2, appelée l'*énergie de courbure* :

$$\mathcal{E}_s \stackrel{\text{def}}{=} \int_{\mathcal{R}} \int_{\mathcal{R}} \left( \frac{\partial \mathcal{W}^2}{\partial^2 x} \right)^2 + 2 \left( \frac{\partial \mathcal{W}^2}{\partial x \partial y} \right)^2 + \left( \frac{\partial \mathcal{W}^2}{\partial^2 y} \right)^2 dx dy. \quad (2.6)$$

En pratique, suivant la fonction de déformation  $\mathcal{W}$  utilisée il peut être nécessaire de discrétiser les intégrales. Le terme de régularisation est alors mis sous la forme suivante :

$$\mathcal{E}_s^2(\mathbf{u}) = \|\mathcal{Z}\mathbf{u}\|^2 = \mathbf{u}^T \mathcal{Y} \mathbf{u} \quad \text{avec} \quad \mathcal{Y} = \mathcal{Z}^T \mathcal{Z}. \quad (2.7)$$



La matrice  $\mathcal{Y}$  possède la structure suivante :  $\mathcal{Y} = \text{diag}(\mathcal{C}, \mathcal{C})$ . Pour le lissage quadratique de l'équation (2.5),  $\mathcal{C} = \mathcal{C}_x^\top \mathcal{C}_x + \mathcal{C}_y^\top \mathcal{C}_y$  avec  $\mathcal{C}_x, \mathcal{C}_y$  des opérateurs de différence finie approximant respectivement les dérivées premières  $\frac{\partial}{\partial x}$  et  $\frac{\partial}{\partial y}$ . Pour l'énergie de courbure,  $\mathcal{C} = \mathcal{C}_{xx}^\top \mathcal{C}_{xx} + 2\mathcal{C}_{xy}^\top \mathcal{C}_{xy} + \mathcal{C}_{yy}^\top \mathcal{C}_{yy}$ , avec  $\mathcal{C}_{xx}, \mathcal{C}_{xy}, \mathcal{C}_{yy}$  sont des opérateurs de différence finie approximant respectivement les dérivées secondes  $\frac{\partial^2}{\partial x^2}, \frac{\partial^2}{\partial x \partial y}$  et  $\frac{\partial^2}{\partial y^2}$ . La forme discrétisée de l'énergie de courbure est utilisée au chapitre 4 pour régulariser un champ de déplacement.

Plus de détails sur le calcul de la matrice  $\mathcal{Y}$  peuvent être trouvés dans (Prasad et al., 2006). Ces termes de régularisation sont assez génériques, ils sont valides pour un très grand nombre de classes d'objets.

#### 2.2.1.4 Robustesse aux données aberrantes

Que ce soit pour les méthodes directes ou basées primitives la présence de données aberrantes est inévitable : erreurs de mises en correspondance, intensités lumineuses erronées de par la présence d'occultations, *etc.* Pour les critères décrits précédemment, la contribution d'un résidu à l'erreur totale est quadratique. Plus un résidu est aberrant et plus son influence est grande. L'estimation des transformations tend alors à s'adapter aux données aberrantes. Une manière simple de résoudre ce problème consiste à limiter l'influence des résidus élevés en incluant un estimateur robuste tel un *M-estimateur*  $\rho$ , dans la fonction de coût. Par exemple, le terme de données (2.2) robustifié est donné par :

$$\mathcal{E}_d(u) \stackrel{\text{def}}{=} \sum_{\mathbf{q} \in \mathcal{R}} \rho(\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) - \mathcal{I}_0(\mathbf{q})). \quad (2.8)$$

Une grande variété de M-estimateurs sont proposés dans la littérature, comme par exemple la fonction de Huber (Hager and Belhumeur, 1998; Lepetit and Fua, 2005) :

$$\rho(t) = \begin{cases} \frac{1}{2}t^2 & \text{si } \|t\| \leq \sigma \\ \sigma\|t\| - \frac{1}{2}\sigma^2 & \text{sinon,} \end{cases}$$

le noyau Lorentzien (Fitzgibbon, 2003)  $\rho(t) = \log\left(1 + \frac{t^2}{\sigma}\right)$ , ou encore la fonction de Tuckey (Lepetit and Fua, 2005) :

$$\rho(t) = \begin{cases} \frac{\sigma^2}{6} \left[ 1 - \left( 1 - \left( \frac{t}{\sigma} \right)^2 \right)^3 \right] & \text{si } \|t\| \leq \sigma \\ \frac{\sigma^2}{6} & \text{sinon.} \end{cases}$$

La figure 2.2 représente les estimateurs robustes décrits ci-dessus. Pour la fonction de Huber la relation entre le résidu et l'erreur est quadratique pour des valeurs inférieures à  $\sigma$ , puis linéaire pour celles supérieures à  $\sigma$ . Pour le noyau Lorentzien, la relation est globalement modifiée. Elle est toujours strictement croissante mais la pente associée aux résidus élevés (par rapport à  $\sigma$ ) est très inférieure à celle associée aux résidus plus faibles, ce qui limite l'influence des données aberrantes. Pour la fonction de Tuckey, l'influence des valeurs supérieures à  $\sigma$  est uniforme.

Le paramètre  $\sigma$  inhérent aux estimateurs robustes joue un rôle prépondérant. Il peut être fixé arbitrairement ou bien évalué au cours du processus de minimisation. Par exemple la *valeur absolue des écarts à la médiane* (MAD<sup>6</sup>) estimée à partir des résidus courants peut être utilisée (Hampel et al., 1986; Malis and Marchand, 2006). Un M-estimateur est notamment utilisé en annexe A pour rendre robuste la reconstruction des déformations d'une surface à partir de nuages de points 3D pouvant présenter des données manquantes et aberrantes.

<sup>6</sup>"Median Absolute Deviation" en anglais.

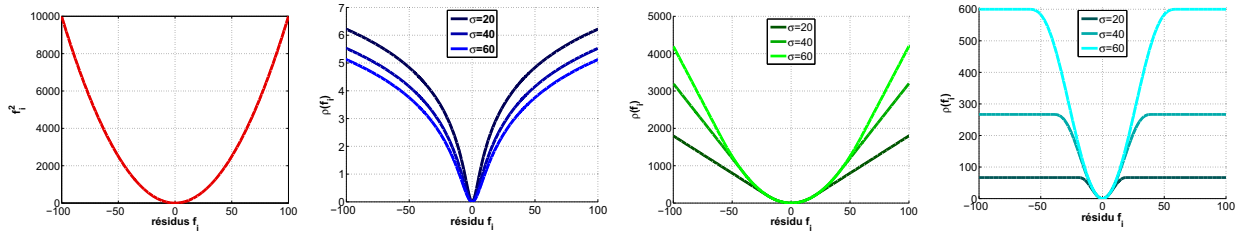


FIG. 2.2 – **Exemples d’estimateurs robustes.** De gauche à droite : la relation quadratique classique, puis le noyau Lorentzien, la fonction de Huber et la fonction de Tuckey pour différentes valeurs de  $\sigma$ .

## 2.2.2 Méthodes directes

### 2.2.2.1 Gestion des variations d’illumination

Le critère de référence (2.2) des méthodes directes est extrêmement sensible aux variations d’illumination. Deux images du même objet prises dans des conditions différentes (caméra, éclairage, *etc.*) peuvent avoir pour chaque pixel des variations d’intensité importantes. Afin de palier ce problème, deux manières de procéder existent. Les variations d’illumination peuvent être explicitement modélisées dans la fonction de coût. Une transformation photométrique entre les images est alors définie. Il en résulte le terme de données suivant :

$$\mathcal{E}_d(u) \stackrel{\text{def}}{=} \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{P}_h(\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})); \mathbf{u}_p) - \mathcal{I}_0(\mathbf{q})\|^2. \quad (2.9)$$

La transformation  $\mathcal{P}_h(\cdot; \mathbf{u}_p)$  représente soit des variations d’illumination globales : gain et biais, modèle affine (Bartoli, 2006), soit locales : spline, modèle affine par morceaux (Silveira and Malis, 2007).

Une autre solution est d’estimer la transformation géométrique sur des données invariantes aux conditions d’éclairage. Par exemple une simple normalisation des images permet d’introduire une certaine robustesse aux changements globaux d’illumination. Elle annule les variations de type gain et biais qui apparaissent notamment lorsque les images sont prises par deux appareils photographiques différents. Dans (Pizarro and Bartoli, 2007), les images sont projetées dans un espace invariant aux phénomènes d’ombrages. Pour cela des paramètres supplémentaires de calibration photométrique doivent être estimés simultanément avec la transformation géométrique.

### 2.2.2.2 Approche multi-résolution

Afin de remédier partiellement à l’aspect local des méthodes directes, une procédure *multi-résolution* est fréquemment utilisée. L’idée est d’estimer la transformation géométrique sur des images de faible résolution puis de propager le champ de déplacement estimé aux images de plus haute résolution. Les étapes supplémentaires introduites par l’approche multi-résolution sont la construction d’une pyramide d’images et la propagation des déplacements estimés le long de la pyramide. A chaque niveau de la pyramide, l’image du niveau inférieur voit ses dimensions divisées par 2, le niveau le plus bas étant l’image d’origine. Différentes approches peuvent être utilisées pour la construction d’une image basse résolution à partir d’une image de plus haute résolution. Une méthode classique consiste à calculer chaque pixel de l’image du niveau  $k$  comme une moyenne pondérée des pixels du niveau  $k - 1$  appartenant à une fenêtre d’intérêt. Nous utilisons par la suite une pondération Gaus-

sienne avec une fenêtre de taille 5 pixels. Cette opération est appelée *réduction*. Enfin, la propagation des champs de déplacement le long de la pyramide se fait par un opérateur d'*expansion* qui est l'opérateur « inverse » de la réduction (Burt and Adelson, 1983). La figure 2.3 illustre le principe de l'approche multi-résolution en recalage d'images. Une procédure multi-résolution est utilisée au chapitre 4 pour permettre l'estimation de déplacement important engendré par l'auto-occultation d'une surface déformable.

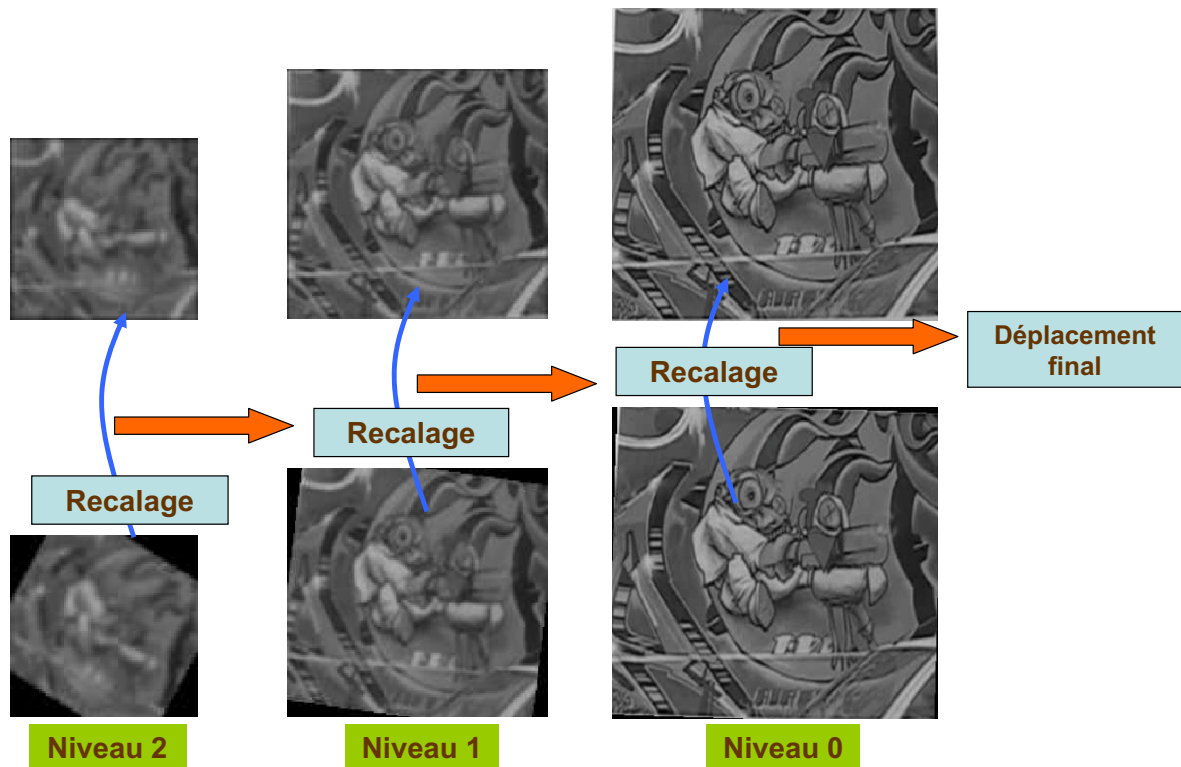


FIG. 2.3 – L'approche multi-résolution en recalage d'images. Le recalage se fait tout d'abord sur les images de faible résolution puis le champ de déplacement estimé est propagé au niveau supérieur. Ces étapes sont répétées jusqu'à ce que le dernier niveau de la pyramide soit atteint.

### 2.2.2.3 Méthodes d'optimisation

**Lois de mise à jour.** Pour minimiser le critère SSD des approches directes donné par l'équation (2.2), deux lois de mise à jour sont possibles : la *mise à jour additive* et la *mise à jour compositionnelle*. La première s'applique dans l'espace des paramètres :  $\mathbf{u} \leftarrow \mathbf{u} + \Delta\mathbf{u}$  tandis que la deuxième se fait dans l'espace des transformations :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \leftarrow \mathcal{W}(\mathbf{q}; \mathbf{u}) \circ \mathcal{W}(\mathbf{q}; \Delta\mathbf{u}). \quad (2.10)$$

Les deux lois de mises à jour sont compatibles avec les différentes approximations de la fonction coût présentées §1.6 comme par exemple Gauss-Newton. Les lois de mise à jour additive et compositionnelle appliquées au critère SSD donnent :

$$\text{Mise à jour additive} \quad \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u} + \Delta \mathbf{u})) - \mathcal{I}_0(\mathbf{q})\|^2 \quad (2.11)$$

$$\text{Mise à jour compositionnelle} \quad \sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{I}(\mathcal{W}(\mathcal{W}(\mathbf{q}; \Delta \mathbf{u}); \mathbf{u})) - \mathcal{I}_0(\mathbf{q})\|^2. \quad (2.12)$$

Pour la loi de mise à jour additive l'incrément des paramètres  $\Delta \mathbf{u}$  est estimé autour des paramètres courant  $\mathbf{u}$  tandis que pour la mise à jour compositionnelle, il est estimé autour des paramètres de référence  $\mathbf{u}_0$ . On parle alors de *recalage locale*.

**Approches compositionnelles directe et inverse.** Elles diffèrent au niveau du « sens » du recalage locale. D'après l'équation 2.12, l'incrément des paramètres  $\Delta \mathbf{u}$  est évalué en recalant localement l'image  $\mathcal{I}_W = \mathcal{I}(\mathcal{W}(\cdot; \mathbf{u}))$  sur l'image de référence  $\mathcal{I}_0$ . On parle de *recalage local direct*. Une autre possibilité est de localement recaler l'image de référence  $\mathcal{I}_0$  sur l'image  $\mathcal{I}_W$  (Baker and Matthews, 2004). On parle alors de *recalage local inverse*. Il en résulte la loi de mise à jour compositionnelle inverse donnée par :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \leftarrow \mathcal{W}(\mathbf{q}; \mathbf{u}) \circ \mathcal{W}(\mathbf{q}; \Delta \mathbf{u})^{-1}. \quad (2.13)$$

La différence avec la mise à jour compositionnelle directe est l'inversion de la transformation locale  $\mathcal{W}(\mathbf{q}; \Delta \mathbf{u})$  avant la composition.

Le critère SSD (2.2) associé à une loi de mise à jour compositionnelle et à un recalage locale inverse est donné par :

$$\sum_{\mathbf{q} \in \mathcal{R}} \|\mathcal{I}_0(\mathcal{W}(\mathbf{q}; \Delta \mathbf{u})) - \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))\|^2. \quad (2.14)$$

D'un point de vue schématique, les algorithmes à mise à jour compositionnelle (directe et inverse) se décomposent en trois étapes :

- ▷ **Étape 1 : Recalage global.** La transformation courante  $\mathcal{W}(\cdot; \mathbf{u})$  est utilisée pour engendrer l'image  $\mathcal{I}_W : \mathcal{I}_W = \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))$ .
- ▷ **Étape 2 : Recalage local.** La transformation locale  $\mathcal{W}(\cdot; \Delta \mathbf{u})$  recalant l'image de référence  $\mathcal{I}_W$  sur  $\mathcal{I}_0$  ou  $\mathcal{I}_0$  sur  $\mathcal{I}_W$  est estimée.
- ▷ **Étape 3 : Mise à jour.** La transformation finale est obtenue par (2.10) ou respectivement par (2.13).

L'avantage de la mise à jour compositionnelle est qu'elle peut introduire, sous certaines conditions, une matrice Jacobienne constante : le recalage local doit être inverse et la fonction de coût approximée par Gauss-Newton. En contrepartie, les opérations de composition et / ou d'inversion font que ces algorithmes ne peuvent être appliqués qu'avec des transformations formant un groupe, par exemples homographies, affinités, *etc.* La figure 2.4 illustre le principe de la loi de mise à jour compositionnelle en comparaison de la loi additive.

Ci-dessous nous décrivons brièvement des exemples algorithmes d'optimisation utilisés pour minimiser le critère (2.2). Ils diffèrent du point de vue de la loi de mise à jour et de l'approximation de la fonction de coût.

**Gauss-Newton additif.** L'algorithme Gauss-Newton additif est très fréquemment utilisé pour minimiser le critère (2.2). Il est souvent préféré à l'algorithme Levenberg-Marquardt additif. Leurs propriétés de convergence sont semblables (Baker and Matthews, 2004).

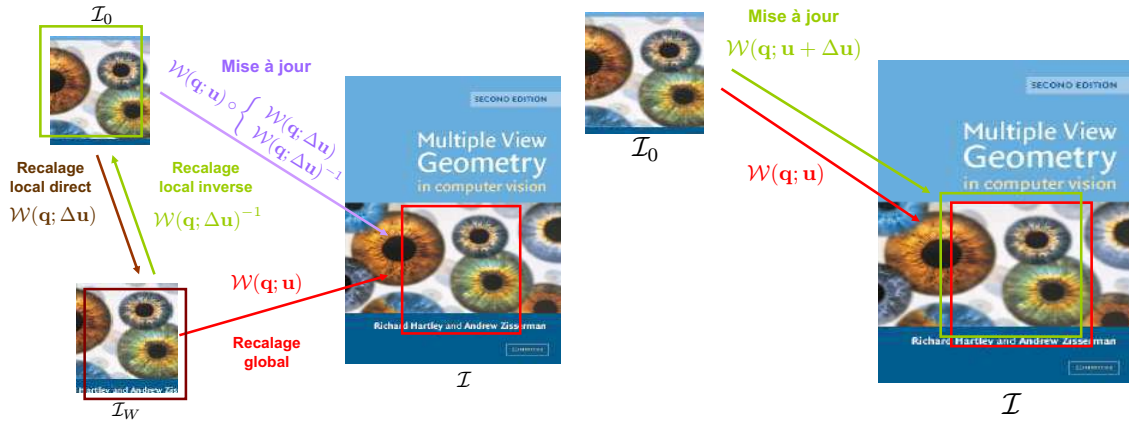


FIG. 2.4 – **Comparaison des lois de mises à jour additive et compositionnelle.** A gauche : la mise à jour compositionnelle, l'estimation de l'incrément  $\Delta \mathbf{u}$  se fait localement autour des paramètres de l'image de référence  $\mathbf{u}_0$ . La mise à jour se fait par composition dans l'espace des transformations. A droite : l'approche additive, l'estimation de l'incrément  $\Delta \mathbf{u}$  se fait autour des paramètres courants  $\mathbf{u}$ , la mise à jour est additive dans l'espace des paramètres.

	Sens direct	Sens inverse
mise à jour additive	DA "Direct Additive"	×
mise à jour compositionnelle	DC "Direct Compositionnal"	IC "Inverse Compositionnal"

TAB. 2.1 – Les différentes terminologies utilisées pour nommer les algorithmes minimisant le critère SSD. Notons qu'un algorithme de type "Inverse Additive" est proposée dans (Hager and Belhumeur, 1998).

L'approximation de Gauss-Newton appliquée à l'équation (2.11) donne :

$$\sum_{\mathbf{q} \in \mathcal{R}} \left\| \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) + \mathbf{g}^T(\mathbf{q}; \mathbf{u}) \Delta \mathbf{u} - \mathcal{I}_0(\mathbf{q}) \right\|^2. \quad (2.15)$$

La matrice Jacobienne est obtenue en rassemblant les vecteurs gradients  $\mathbf{g}^T(\mathbf{q}; \mathbf{u})$ . Ces derniers sont donnés par le produit entre le gradient de l'image courante  $\mathcal{I}$  évaluée en  $\mathcal{W}(\mathbf{q}; \mathbf{u})$  et le gradient de la transformation  $\mathcal{W}$  évaluée aux paramètres courants  $\mathbf{u}$  et au pixel  $\mathbf{q}$  :

$$\mathbf{g}^T(\mathbf{q}; \mathbf{u}) = \nabla \mathcal{I}^T|_{\mathcal{W}(\mathbf{q}; \mathbf{u})} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}|_{(\mathbf{q}; \mathbf{u})}.$$

Nous appelons cet algorithme DA-GN pour "Direct Additive Gauss-Newton". La première lettre est relative au sens du recalage (relatif aux rôles de l'image de référence et de l'image courante dans la fonction de coût) et la deuxième au type de mise à jour. Le tableau 2.1 récapitule les différentes terminologies utilisées en fonction de la loi de mise à jour et du sens du recalage.

Le tableau 2.2 récapitule les étapes de l'algorithme DA-GN. Les performances de cet algorithme sont comparées, au chapitre 3, avec celles des algorithmes de recalage d'images *guidés par primitives* que nous proposons. Il est également utilisé au chapitre 4 pour minimiser la fonction de coût proposée pour la gestion des auto-occultations en recalage d'images.

**Efficient Second Order Minimisation additif.** L'approximation de l'équation (2.11) par ESM donne :

**En ligne**

- ▷ Recaler par  $\mathcal{W}(\mathbf{q}; \mathbf{u})$  l'image courante sur l'image de référence :  $\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))$ .
- ▷ Calculer l'image de différence  $\mathcal{D}(\mathbf{q}) = \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) - \mathcal{I}_0(\mathbf{q})$ .
- ▷ Évaluer le gradient de l'image courante  $\nabla \mathcal{I}$  à  $\mathcal{W}(\mathbf{q}; \mathbf{u})$ .
- ▷ Évaluer le gradient de la transformation  $\frac{\partial \mathcal{W}}{\partial \mathbf{u}}$  à  $(\mathbf{q}; \mathbf{u})$ .
- ▷ Calculer les vecteurs gradients  $\mathbf{g}^\top(\mathbf{q}; \mathbf{u}) = \nabla \mathcal{I}^\top|_{\mathcal{W}(\mathbf{q}; \mathbf{u})} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}|_{(\mathbf{q}; \mathbf{u})}$ .
- ▷ Former la matrice Jacobienne  $\mathcal{J}$  en rassemblant les vecteurs gradients  $\mathbf{g}^\top(\mathbf{q}; \mathbf{u})$ .
- ▷ Calculer l'approximation de Gauss-Newton de la matrice Hessienne  $\mathbf{H} = \mathcal{J}^\top \mathcal{J}$ .
- ▷ Calculer  $\mathbf{b} = \mathcal{J}^\top \mathcal{D}$ .
- ▷ Calculer  $\Delta \mathbf{u} = -\mathbf{H}^{-1} \mathcal{J}^\top \mathbf{b}$ .
- ▷ Mettre à jour les paramètres courants  $\mathbf{u} \leftarrow \mathbf{u} + \Delta \mathbf{u}$ .

TAB. 2.2 – Les différentes étapes d'une itération de l'algorithme DA-GN.

$$\sum_{\mathbf{q} \in \mathcal{R}} \left\| \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) + \frac{1}{2} (\mathbf{g}^\top(\mathbf{q}; \mathbf{u}) + \mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0)) \Delta \mathbf{u} - \mathcal{I}_0(\mathbf{q}) \right\|^2,$$

Avec  $\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0)$  les vecteurs gradients évalués au niveau de l'image de référence. Ils sont obtenus par le produit entre le gradient de l'image de référence  $\mathcal{I}_0$  évalué en  $\mathcal{W}(\mathbf{q}; \mathbf{u}_0)$  et le gradient de la transformation  $\mathcal{W}$  évalué à  $(\mathbf{q}; \mathbf{u}_0)$  :  $\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0) = \nabla \mathcal{I}_0^\top|_{\mathcal{W}(\mathbf{q}; \mathbf{u}_0)} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}|_{(\mathbf{q}; \mathbf{u}_0)}$ . Avec  $\mathbf{u}_0$  les paramètres de la transformation identité :  $\mathcal{W}(\mathbf{q}; \mathbf{u}_0) = \mathbf{q}$ .

La matrice Jacobienne introduite n'est pas constante puisque  $\mathbf{g}^\top(\mathbf{q}; \mathbf{u})$  dépend des paramètres courants, l'une de ses composantes peut être néanmoins estimée hors ligne. Dans la suite du mémoire, afin d'être consistant avec nos notations, nous appelons cet algorithme DA-ESM pour "Direct Additive Efficient Second order Minimisation". Notons que l'approximation de ESM est combinée avec une loi de mise à jour compositionnelle dans (Benhimane and Malis, 2004; Silveira and Malis, 2007). Il en résulte l'algorithme DC-ESM "Direct Compositionnel Efficient Second order Minimisation". Le tableau 2.3 récapitule les étapes de l'algorithme DA-ESM. Les performances de cet algorithme sont comparées, au chapitre 3, avec celles des algorithmes de recalage d'images guidés par primitives que nous proposons.

**Gauss-Newton compositionnel inverse.** En appliquant l'approximation de Gauss-Newton à (2.14) on obtient :

$$\sum_{\mathbf{q} \in \mathcal{R}} \left\| \mathcal{I}_0(\mathcal{W}(\mathbf{q}; \mathbf{u}_0)) + \mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0) \Delta \mathbf{u} - \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) \right\|^2. \quad (2.16)$$

La matrice Jacobienne  $\mathcal{J}_0$ , obtenue en rassemblant les vecteurs gradients  $\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0)$ , ne dépend plus des paramètres courants  $\mathbf{u}$  : elle est constante et peut donc être calculée hors ligne. La variation locale des paramètres est donnée par :

$$\Delta \mathbf{u} = -\mathcal{J}_0^\dagger (\mathcal{I}_0(\mathbf{q}) - \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))).$$

Nous appelons cet algorithme IC-GN pour "Inverse Compositionnal Gauss-Newton". Ses différentes étapes sont récapitulées sur le tableau 2.4.

<b>Hors ligne</b>
<ul style="list-style-type: none"> <li>▷ Évaluer le gradient de l'image de référence <math>\nabla \mathcal{I}_0</math> à <math>\mathcal{W}(\mathbf{q}; \mathbf{u}_0) = \mathbf{q}</math>.</li> <li>▷ Évaluer le gradient de la transformation <math>\frac{\partial \mathcal{W}}{\partial \mathbf{u}}</math> à <math>(\mathbf{q}; \mathbf{u}_0)</math>.</li> <li>▷ Calculer les vecteurs gradients <math>\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0) = \nabla \mathcal{I}_0^\top  _{\mathcal{W}(\mathbf{q}; \mathbf{u}_0)} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}  _{(\mathbf{q}; \mathbf{u}_0)}</math>.</li> </ul>
<b>En ligne</b>
<ul style="list-style-type: none"> <li>▷ Recaler par <math>\mathcal{W}(\mathbf{q}; \mathbf{u})</math> l'image courante sur l'image de référence : <math>\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))</math>.</li> <li>▷ Calculer l'image de différence <math>\mathcal{D}(\mathbf{q}) = \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})) - \mathcal{I}_0(\mathbf{q})</math>.</li> <li>▷ Évaluer le gradient de l'image courante <math>\nabla \mathcal{I}</math> à <math>\mathcal{W}(\mathbf{q}; \mathbf{u})</math>.</li> <li>▷ Évaluer le gradient de la transformation <math>\frac{\partial \mathcal{W}}{\partial \mathbf{u}}</math> à <math>(\mathbf{q}; \mathbf{u})</math>.</li> <li>▷ Calculer les vecteurs gradients <math>\mathbf{g}^\top(\mathbf{q}; \mathbf{u}) = \nabla \mathcal{I}^\top  _{\mathcal{W}(\mathbf{q}; \mathbf{u})} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}  _{(\mathbf{q}; \mathbf{u})}</math>.</li> <li>▷ Former la matrice Jacobienne <math>\mathcal{J}</math> en rassemblant les vecteurs gradients <math>\frac{1}{2}(\mathbf{g}^\top(\mathbf{q}; \mathbf{u}) + \mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0))</math>.</li> <li>▷ Calculer l'incrément <math>\Delta \mathbf{u} = -\mathcal{J}^\dagger \mathcal{D}</math>.</li> <li>▷ Mettre à jour les paramètres courants <math>\mathbf{u} \leftarrow \mathbf{u} + \Delta \mathbf{u}</math>.</li> </ul>

TAB. 2.3 – Les différentes étapes d'une itération de l'algorithme DA-ESM.

<b>Hors ligne</b>
<ul style="list-style-type: none"> <li>▷ Évaluer le gradient de l'image de référence <math>\nabla \mathcal{I}_0</math> à <math>\mathcal{W}(\mathbf{q}; \mathbf{u}_0) = \mathbf{q}</math>.</li> <li>▷ Évaluer le gradient de la transformation <math>\frac{\partial \mathcal{W}}{\partial \mathbf{u}}</math> à <math>(\mathbf{q}; \mathbf{u}_0)</math>.</li> <li>▷ Calculer les vecteurs gradients <math>\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0) = \nabla \mathcal{I}_0^\top  _{\mathcal{W}(\mathbf{q}; \mathbf{u}_0)} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}  _{(\mathbf{q}; \mathbf{u}_0)}</math>.</li> <li>▷ Former la matrice Jacobienne <math>\mathcal{J}_0</math> en rassemblant les vecteurs gradients <math>\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0)</math>.</li> <li>▷ Calculer la pseudo inverse de <math>\mathcal{J}_0</math> : <math>\mathcal{J}_0^\dagger</math>.</li> </ul>
<b>En ligne</b>
<ul style="list-style-type: none"> <li>▷ Recaler par <math>\mathcal{W}(\mathbf{q}; \mathbf{u})</math> l'image courante sur l'image de référence : <math>\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))</math>.</li> <li>▷ Calculer l'image de différence <math>\mathcal{D}(\mathbf{q}) = \mathcal{I}_0(\mathbf{q}) - \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))</math>.</li> <li>▷ Calculer l'incrément <math>\Delta \mathbf{u} = -\mathcal{J}_0^\dagger \mathcal{D}</math>.</li> <li>▷ Mettre à jour la transformation courante : <math>\mathcal{W}(\mathbf{q}; \mathbf{u}) \leftarrow \mathcal{W}(\mathbf{q}; \mathbf{u}) \circ \mathcal{W}(\mathbf{q}; \Delta \mathbf{u})^{-1}</math>.</li> </ul>

TAB. 2.4 – Les différentes étapes d'une itération de l'algorithme IC-GN.

### 2.2.3 Méthodes basées primitives

Elles s'articulent principalement autour de trois grandes étapes : l'extraction de primitives (généralement des points d'intérêt) indépendamment dans chaque image, leur mise en correspondance et l'estimation robuste de la transformation. La figure 2.5(b) résume le principe des méthodes basées primitives.

**Extraction de points d'intérêt.** Un *point d'intérêt* correspond en général à un point de l'espace détectable de manière robuste au changement de point de vue ou d'éclairage. Il en existe plusieurs types. Les plus courants sont les « coins », où le gradient de l'intensité lumineuse est fort dans deux directions. Ces points sont facilement identifiables dans les images car ils correspondent souvent aux coins des objets de la scène ou aux détails de texture. Un grand nombre de méthodes existent pour extraire des points d'intérêt d'une image. Une comparaison des détecteurs existants est réalisée dans (Mikolajczyk and Schmid, 2005). Plus récemment, le détecteur FAST (Rosten and Drummond, 2005) a été proposé.

**La mise en correspondance.** L'approche « classique » de mise en correspondance consiste à extraire et à comparer les signatures locales, appelées *descripteurs*, associées à chaque point d'intérêt. Un descripteur local permet de résumer l'information contenue dans le voisinage du point d'intérêt. Une fois les descripteurs extraits, la mise en correspondance se fait par mesure de similarité (notion de distance) entre les descripteurs.

La qualité d'un descripteur se juge par son invariance aux transformations géométrique et photométrique. L'un des plus performants est le descripteur SIFT ("Scale Invariant Feature Transform") introduit dans (Lowe, 2004). Ce descripteur est basé sur une approche multi-résolution et sur la construction d'histogrammes de l'orientation des gradients autour du point d'intérêt. Notons que le descripteur SURF ("Speeded Up Robust Features") (Bay et al., 2006), plus récent s'avère être également très performant.

Dans (Lepetit et al., 2004), le problème de mise en correspondance est traité comme un problème de classification. Une classe étant l'ensemble des vues possibles d'un point d'intérêt. Le *classificateur* est appris hors ligne sur l'ensemble des vues (générées synthétiquement) associées à chaque point d'intérêt. Des classificateurs de type "randomized trees" (Lepetit et al., 2005) et "ferns" (Ozuysal et al., 2007) ont été testés. Ils présentent un taux de mises en correspondance exactes remarquable.

**Estimation robuste de la transformation.** Cette dernière étape revient à minimiser l'équation (2.4), introduite en §2.2.1.2, par moindres carrés non-linéaire. Les mises en correspondance étant sujettes à erreur, des méthodes robustes doivent être utilisées. Par exemple inclure un M-estimateur, décrit §2.2.1.4, pour robustifier la fonction de coût. Une autre possibilité est l'algorithme RANSAC ("RANdom SAMple Consensus") (Fischler and Bolles, 1981). Il fonctionne par tirage aléatoire afin d'éliminer les valeurs aberrantes dans les correspondances de points. Néanmoins, pour les modèles déformables, utiliser l'algorithme RANSAC peut s'avérer être coûteux en temps de calcul en raison du nombre important de paramètres.

Dans l'équation (2.4), le poids  $w_c$  contrôle l'importance donnée à une paire de points mise en correspondance. Il permet notamment de résoudre le problème des mises en correspondance multiples. Ce poids peut être fixé de différentes manières, comme par exemple :  $w_c = 1$  pour la correspondance la plus proche et  $w_c = 0$  pour les autres comme pour l'algorithme "Itérative Closest Point" (ICP). Où



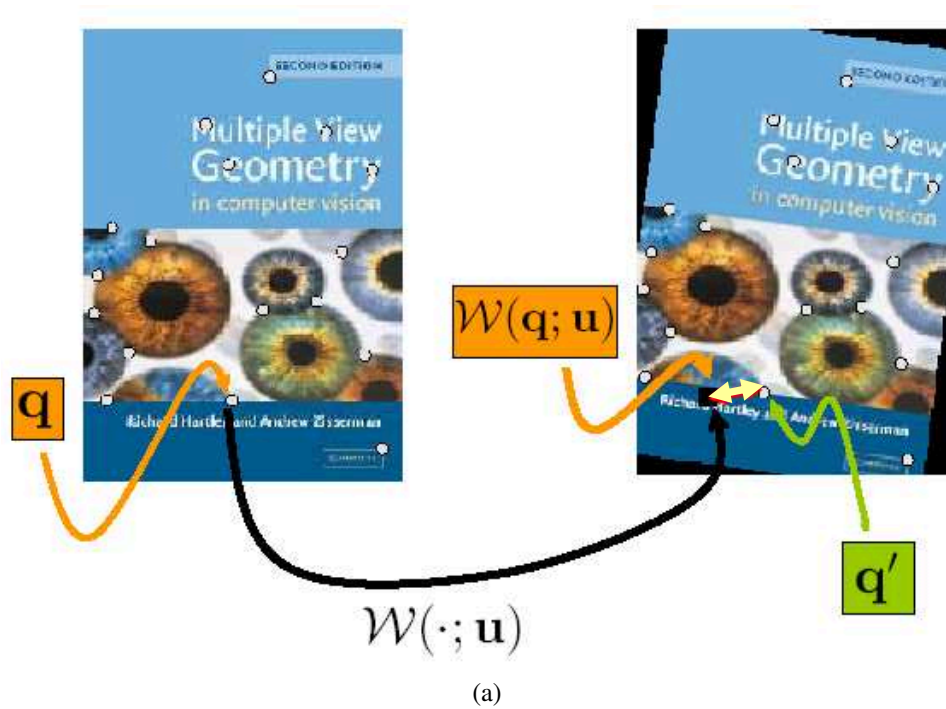


FIG. 2.5 – Principe des méthodes basées primitives de recalage d'images. (a) Illustration du critère des méthodes basées primitives. (b) Schéma récapitulatif de leur fonctionnement.

encore  $w_c = \frac{\exp(-\|\mathbf{q}' - \mathcal{W}(\mathbf{q}; \mathbf{u})\|^2 / 2\sigma^2)}{\sum_{\mathbf{q} \leftrightarrow \mathbf{q}' \in \Theta} \exp(-\|\mathbf{q}' - \mathcal{W}(\mathbf{q}; \mathbf{u})\|^2 / 2\sigma^2)}$  comme pour l'algorithme EM<sup>7</sup>-ICP (Granger and Penec, 2002). Dans (Pilet et al., 2008), les auteurs montrent que l'estimateur robuste a une influence prépondérante sur les propriétés de convergence par rapport aux poids  $w_c$ . Ils fixent ces derniers à 1 pour toutes les correspondances sans dégrader la précision de la transformation estimée.

## 2.3 Modélisation des déformations image

Dans la section précédente, nous avons vu les outils de base permettant d'estimer les déformations entre les images. Cette section se consacre à la modélisation des déformations image par une fonction de déformation. Elle décrit deux grandes classes de fonctions de déformation utilisées dans notre étude.

### 2.3.1 Les fonctions à base radiale

#### 2.3.1.1 Définition

Sous leur forme originelle, les *fonctions à base radiale* (RBFs<sup>8</sup>) définissent une transformation de  $\mathbb{R}^d$  dans  $\mathbb{R}$ , où  $d$  est la dimension de l'espace d'origine. Elles sont définies par des *fonctions noyau*  $\phi$  dont l'argument est la distance Euclidienne entre un point et un centre de la transformation. Une grande variété de fonction noyau est disponible dans la littérature, présentant différentes propriétés. Les plus utilisées sont les fonctions Gaussienne, multiquadrique et plaque mince. Elles sont décrites plus en détails en §2.3.1.3. Une RBF de  $\mathbb{R}^2$  dans  $\mathbb{R}$  est définie par l'équation suivante :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \eta q^x + \mu q^y + \nu + \sum_{k=1}^p \omega_k \phi(d^2(\mathbf{q}, \mathbf{c}_k)). \quad (2.17)$$

Elle est composée de deux parties : une partie linéaire définie par les coefficients  $(\eta \ \mu \ \nu)$  et une combinaison linéaire pondérée de  $p$  fonctions noyau  $\phi$ , avec  $\omega_k$  les coefficients de pondération. Ces fonctions sont appliquées à la distance Euclidienne entre  $\mathbf{q}$  et les centres  $\mathbf{c}_k$  de la transformation. Ces derniers sont placés arbitrairement dans la région d'intérêt. Le vecteur de paramètres  $\mathbf{u}$  concatène l'ensemble des coefficients :  $\mathbf{u}^\top = (\boldsymbol{\omega}^\top \ \eta \ \mu \ \nu)$ .

Cette équation peut être réécrite de la manière suivante :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \mathbf{o}_\mathbf{q}^\top \mathbf{u}, \quad (2.18)$$

avec  $\mathbf{o}_\mathbf{q}^\top = (\phi(d^2(\mathbf{q}, \mathbf{c}_1)) \ \cdots \ \phi(d^2(\mathbf{q}, \mathbf{c}_p)) \ \mathbf{q}^\top \ 1)$ .

Dans notre étude, nous utilisons principalement des transformations basées image de  $\mathbb{R}^2$  dans  $\mathbb{R}^2$ . Elles sont construites à partir de deux RBFs de  $\mathbb{R}^2$  dans  $\mathbb{R}$  :  $\mathcal{W}^x$  et  $\mathcal{W}^y$  partageant les mêmes centres  $\{\mathbf{c}_k\}_{k=1}^p$  :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \begin{pmatrix} \mathcal{W}^x(\mathbf{q}; \mathbf{u}_x) \\ \mathcal{W}^y(\mathbf{q}; \mathbf{u}_y) \end{pmatrix} = \underbrace{\begin{pmatrix} \eta^x & \mu^x & \nu^x \\ \eta^y & \mu^y & \nu^y \end{pmatrix}}_A \mathbf{q} + \sum_{k=1}^p \begin{pmatrix} \omega_k^y \\ \omega_k^x \end{pmatrix} \phi(d^2(\mathbf{q}, \mathbf{c}_k)) = \mathbf{o}_\mathbf{q}^\top (\mathbf{u}_x \ \mathbf{u}_y), \quad (2.19)$$

<sup>7</sup>"Expectation-Maximization" en anglais.

<sup>8</sup>"Radial Basis Functions" en anglais.

où  $\mathcal{A}$  représente une transformation Affine,  $\mathbf{u}_x^\top = (\omega^{x^\top} \ \eta^x \ \mu^x \ \nu^x)$  et  $\mathbf{u}_y^\top = (\omega^{y^\top} \ \eta^y \ \mu^y \ \nu^y)$ . Les poids de la transformation sont rassemblés dans la matrice  $\Omega = (\omega^x \ \omega^y)$  et le vecteur de paramètres est donné par  $\mathbf{u}^\top = (\mathbf{u}_x^\top \ \mathbf{u}_y^\top)$ . Notons que pour certaine fonction noyau  $\phi$ , la présence de la composante polynomiale  $\mathcal{A}$  n'est pas requise, plus de détails sont données en §2.3.1.3.

### 2.3.1.2 Estimation des coefficients

L'estimation des coefficients des RBFs de  $\mathbb{R}^2$  dans  $\mathbb{R}^2$  peut se faire à partir de correspondances de points  $\mathbf{q}_j \leftrightarrow \mathbf{q}'_j$ . Sous la contrainte  $\mathcal{W}(\mathbf{q}_j; \mathbf{u}) = \mathbf{q}'_j$  et en prenant comme centres de la transformation les points  $\{\mathbf{c}_j\}_{j=1}^p : \mathbf{c}_j \leftarrow \mathbf{q}_j$ , l'équation (2.19) devient :

$$\mathbf{c}'_j = \mathcal{W}(\mathbf{c}_j; \mathbf{u}) = \mathcal{A}\mathbf{c}_j + \sum_{k=1}^p \begin{pmatrix} \omega_k^y \\ \omega_k^x \end{pmatrix} \phi(d^2(\mathbf{c}_j, \mathbf{c}_k)). \quad (2.20)$$

En rassemblant les équations (2.20) introduites par chaque centre  $\mathbf{c}_j$  on obtient le système linéaire suivant :

$$\underbrace{\begin{pmatrix} \mathcal{K} & \mathcal{P} \\ \mathcal{P}^\top & \mathbf{0}_{3 \times 3} \end{pmatrix}}_{\text{matrice de transfert}} \begin{pmatrix} \Omega \\ \mathcal{A}^\top \end{pmatrix} = \begin{pmatrix} \mathcal{P}' \\ \mathbf{0}_{3 \times 2} \end{pmatrix}, \quad (2.21)$$

où  $\mathcal{K}_{i,j} = \phi(d^2(\mathbf{c}_i, \mathbf{c}_j))$ , les  $j$ ème lignes de  $\mathcal{P}$  et  $\mathcal{P}'$  sont respectivement données par  $(c_j^x \ c_j^y \ 1)$  et  $(c_j'^x \ c_j'^y \ 1)$ . Les coefficients des RBFs sont estimés en résolvant le système (2.21).

Les trois dernières lignes du système sont les *conditions aux bords*<sup>9</sup>. Elles permettent d'assurer que les équations possèdent des dérivées secondes intégrables deux fois et régularisent la fonction à l'infini. En d'autres termes, les conditions aux bords imposent aux déformations de tendre vers une affinité lorsque l'on s'écarte de la région d'intérêt. Elles sont données par :

$$\left( \sum_{k=1}^p \omega_k^x = \sum_{k=1}^p \omega_k^y = \sum_{k=1}^p \omega_k^x q_k^x = \sum_{k=1}^p \omega_k^y q_k^y = 0 \right) \Leftrightarrow \mathcal{P}^\top \Omega = \mathbf{0}_{3 \times 2}. \quad (2.22)$$

Il est également possible de relaxer la contrainte de mise en correspondance exacte :  $\mathcal{W}(\mathbf{q}_j; \mathbf{u}) \approx \mathbf{q}'_j$ . Pour cela un coefficient de régularisation  $\lambda$  est ajouté à la matrice  $\mathcal{K}$  lors de l'estimation des paramètres :  $\mathcal{K} \leftarrow \mathcal{K} + \lambda \mathbf{I}$ . Quand  $\lambda$  devient grand la transformation se rapproche d'une affinité.

### 2.3.1.3 Propriétés des fonctions à base radiale

Le choix de la fonction noyau  $\phi$  détermine les caractéristiques de la fonction de déformation  $\mathcal{W}$ . Les principales propriétés des RBFs sont :

- ▷ **L'influence spatiale.** Elle peut être locale ou globale. Une *influence locale* signifie qu'un point se situant à une distance d'un centre supérieure à un certain seuil, n'est pas affecté par la position de ce dernier. Seuls les points se situant au voisinage d'un centre voient leurs positions dépendre de celle du centre. Il en résulte que la matrice  $\mathcal{K}$  est creuse. Une *influence globale*,

<sup>9</sup>"Side conditions" en anglais.

quant-à-elle, signifie qu'un centre agit sur l'ensemble de la région d'intérêt. La matrice  $\mathcal{K}$  introduite est pleine.

- ▷ **Temps de calcul.** Le temps requis pour évaluer la fonction de déformation dépend fortement du noyau utilisé. En effet, si la matrice  $\mathcal{K}$  est creuse, l'équation (2.21) peut être résolue plus efficacement.
- ▷ **Singularité.** Pour pouvoir estimer les paramètres de la fonction de déformation, il est nécessaire que la matrice  $\mathcal{K}$  soit non-singulière. Cette contrainte est respectée si les points de données ne sont pas colinéaires et si la fonction de noyau  $\phi$  est conditionnellement définie positive (Fornet et al., 1999). Cependant, le polynôme de degré 1 correspondant à la composante affine de la fonction de déformation permet de garantir sous certaines conditions la non singularité de la matrice de transfert même si les conditions décrites ci-dessus ne sont pas respectées.

**Les fonctions noyau.** Une grande variété de fonctions noyau existe dans la littérature, les plus fréquemment utilisées sont :

**Le noyau Gaussien.**  $\phi_G(a) = \exp\left(-\frac{a^2}{2\sigma^2}\right)$ . La figure 2.6(a) représente l'évolution de  $\phi_G(a)$  en fonction de  $a$ . Le noyau Gaussien (Arad and Reissfeld, 1995) a une influence globale négligeable pour  $a > 3\sigma$ . Le paramètre  $\sigma$  contrôle l'influence spatiale du noyau : plus  $\sigma$  est grand, plus le noyau a une large influence spatiale. La matrice  $\mathcal{K}$  associée est dense puisque le support du noyau Gaussien n'est pas compact. Afin de palier à cet inconvénient, des noyaux semblables ayant un support compact sont proposés dans la littérature : les fonctions de Wendland (Fornet et al., 1999) ou celles dites de transition (Arad and Reissfeld, 1995). Notons que le noyau Gaussien est conditionnellement défini positif : la transformation peut être estimée sans composante polynomiale.

**Les noyaux multiquadrique et inverse multiquadrique.**  $\phi_{MQ}(a) = (a^2 + \sigma^2)^\mu$  et  $\phi_{IMQ}(a) = (a^2 + \sigma^2)^{-\mu}$  avec  $\mu \in \mathcal{R}^+$ . Les figures 2.7(a) et 2.7(b) représentent respectivement l'évolution de  $\phi_{MQ}(a)$  en fonction de  $a$  et de  $\sigma$ , et en fonction de  $a$  et de  $\mu$ . Les figures 2.8(a) et 2.8(b) représentent respectivement l'évolution de  $\phi_{IMQ}(a)$  en fonction de  $a$  et de  $\sigma$ , et en fonction de  $a$  et de  $\mu$ . Le noyau multiquadrique et inverse multiquadrique ont une influence globale. Cependant le noyau inverse multiquadrique a une influence négligeable pour de grandes valeurs de  $a$ . Le paramètre  $\mu$  contrôle l'influence spatiale du centre tandis que  $\sigma$  modifie la valeur du noyau au voisinage de ce dernier. Le noyau inverse multiquadrique est défini positif tandis que le noyau multiquadrique ne l'est pas. Pour ce dernier, le degré minimal de la partie polynomiale assurant la non singularité de la matrice de transfert dépend du paramètre  $\mu$ . Le degré minimal est donné par  $p = \lfloor \mu \rfloor - 1$ , le coefficient  $\mu$  ne pourra donc pas être supérieur à 2 :  $\mu \leq 2$ . Plus de détails sur ces fonctions noyau peuvent être trouvés dans (Ruprecht and Muller, 1993).

**Le noyau plaque mince.** Il a une influence spatiale globale et possède la propriété d'introduire une transformation minimisant l'énergie de courbure. En d'autres termes, les déformations engendrées sont naturellement lisses. Ceci en fait l'un des noyaux les plus fréquemment utilisés dans la littérature. Le noyau plaque mince est donné par :

$$\phi_{TPS}(a) = \begin{cases} \frac{a^{(4-p)} \log(a)}{a^\sigma} & 4 - p \in 2\mathbb{N} \\ \frac{a^{(4-p)}}{\sigma} & \text{sinon.} \end{cases}$$

La figure 2.6(b) représente l'évolution de  $\phi_{TPS}(a)$  en fonction de  $a$  pour  $p = 2$ . Le noyau plaque mince (Bookstein, 1989) n'est pas conditionnellement défini positif. Cependant l'ajout du polynôme de degré 1 permet d'assurer la non-singularité de la matrice de transfert (pour un noyau plaque mince de degré 2).

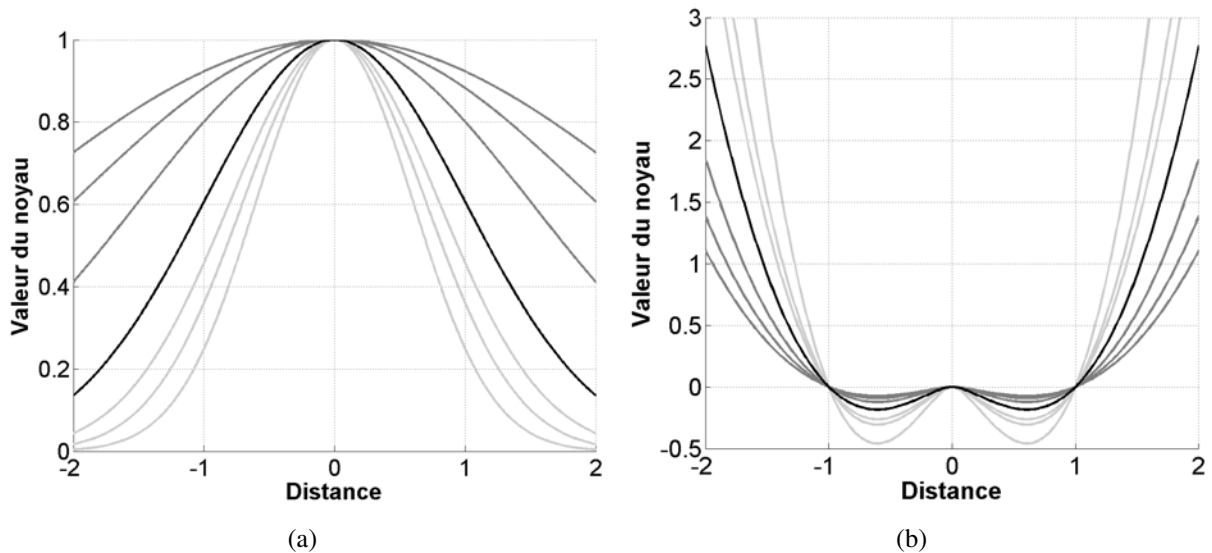


FIG. 2.6 – **Exemples de fonctions noyau pour des transformations RBFs.** (a) Valeur du noyau  $\phi_G(a)$  en fonction de la distance  $a$ . (b) Valeur du noyau  $\phi_{TPS}(a)$  en fonction de la distance  $a$ . Les différentes courbes tracées représentent différentes valeurs du paramètre  $\sigma$ . Les courbes noires correspondent à  $\sigma = 1$ , les courbes gris foncé et gris clair respectivement à  $\sigma > 1$  et  $\sigma < 1$ . Le paramètre  $\sigma$  fixe le degré de l'influence spatiale du noyau.

**Illustration de déformations.** Afin d'illustrer les déformations engendrées par les différentes fonctions noyau, un exemple synthétique est généré. La fonction de déformation est estimée à partir de correspondances de points en suivant le principe présenté en §2.3.1.2. Les différents noyaux, décrits ci-dessus, sont testés.

L'exemple synthétique, représenté sur la figure 2.9(a) consiste à appliquer une rotation de 45 degrés (dans le sens direct), à un carré de dimension  $2 \times 2$  d'une grille régulière de taille  $10 \times 10$ . Les coins de la grille sont, quant à eux, fixes. Les centres de la transformation sont les 8 points constituant le carré et les 4 coins de la grille.

La figure 2.9(b) montre les déformations estimées par des fonctions à base radiale avec les noyaux  $:\phi_G, \phi_{IMQ}, \phi_{MQ}$  et  $\phi_{TPS}$ . Pour les noyaux Gaussien et plaque mince, le coefficient  $\sigma$  est fixé respectivement à  $\sigma = 2$  et  $\sigma = 1$ . Pour le noyau multiquadrique, les coefficients  $\sigma$  et  $\mu$  sont respectivement fixés à  $\sigma = 4$  et  $\mu = 1$  tandis que pour le noyau inverse multiquadrique  $\sigma = 4$  et  $\mu = 3$ . Les RBFs avec un noyau à influence spatiale limitée ( $\phi_G, \phi_{IMQ}$ ), c'est-à-dire pour lesquelles la valeur du noyau diminue avec la distance, ne déforment que la région mise en mouvement ainsi que son voisinage proche tandis que celles présentant un noyau dont la valeur augmente avec la distance ( $\phi_{TPS}, \phi_{MQ}$ ) déforment l'ensemble de la région d'intérêt. Le choix d'un noyau dépend principalement de l'application visée. En imagerie médicale des noyaux locaux ou à influence limitée sont fréquemment privilégiés par exemple pour l'étude des tumeurs (Fornefett et al., 1999). Le noyau de

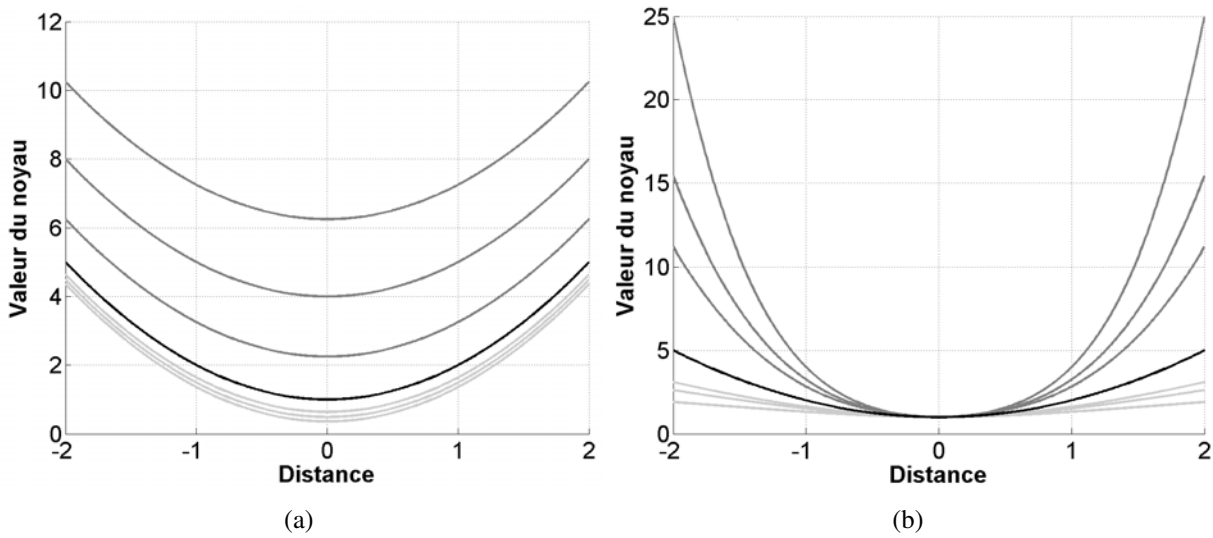


FIG. 2.7 – Exemples de fonctions noyau pour des transformations RBFs. (a) Valeur du noyau  $\phi_{MQ}(a)$  en fonction de la distance  $a$  pour différentes valeurs de  $\sigma$  avec  $\mu = 1$ . (b) Valeur du noyau  $\phi_{MQ}(a)$  en fonction de la distance  $a$  pour différentes valeurs de  $\mu$  avec  $\sigma = 1$ . Les courbes noires correspondent respectivement à  $\sigma = 1$  (a) et  $\mu = 1$  (b), les courbes gris foncé à  $\sigma > 1$  et  $\mu > 1$  et les courbes gris clair à  $\sigma < 1$  et  $\mu < 1$ . Le paramètre  $\sigma$  influe globalement sur la valeur du noyau tandis que  $\mu$  joue le rôle d'un exposant quelconque.

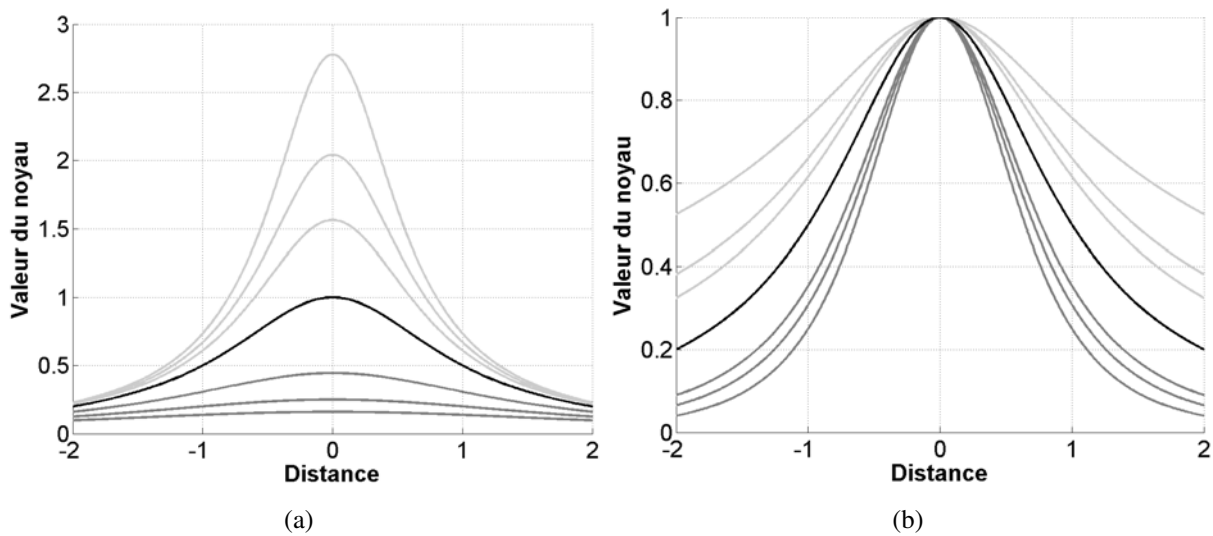


FIG. 2.8 – Exemples de fonctions noyau pour des transformations RBFs. (a) Valeur du noyau  $\phi_{IMQ}(a)$  en fonction de la distance  $a$  pour différentes valeurs de  $\sigma$  avec  $\mu = 1$ . (b) Valeur du noyau  $\phi_{IMQ}(a)$  en fonction de la distance  $a$  pour différentes valeurs de  $\mu$  avec  $\sigma = 1$ . Les courbes noires correspondent respectivement à  $\sigma = 1$  (a) et  $\mu = 1$  (b), les courbes gris foncé à  $\sigma > 1$  et  $\mu > 1$  et les courbes gris clair à  $\sigma < 1$  et  $\mu < 1$ . Le paramètre  $\sigma$  influe sur la valeur du noyau pour des points proche du centre tandis que  $\mu$  contrôle la largeur de sa zone d'influence.

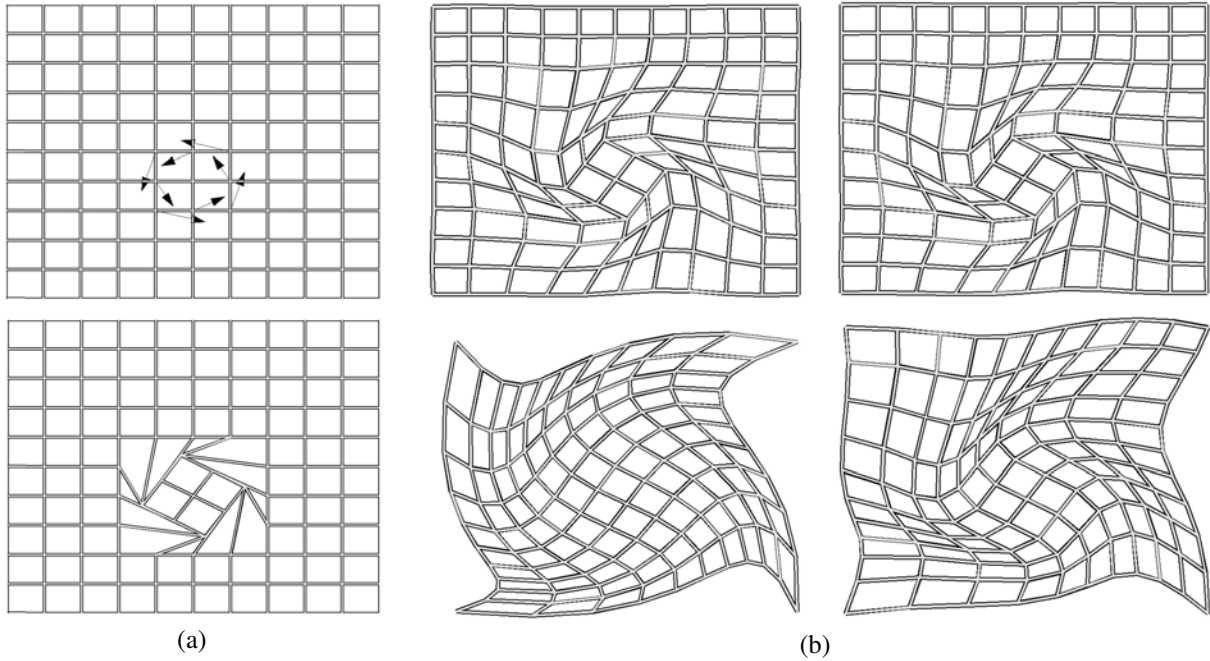


FIG. 2.9 – Exemples de déformations par RBFs pour différentes fonctions noyau. (a) Exemple synthétique généré : rotation d'une région d'intérêt de taille  $2 \times 2$  d'une grille dont les coins sont fixés. (b) Déformations par RBFs pour différentes fonctions noyau. En haut à gauche déformations obtenues avec le noyau  $\phi_G$  ( $\sigma = 1$ ), en haut à droite avec le noyau  $\phi_{IMQ}$  ( $\sigma = 4$ ,  $\mu = 3$ ), en bas à gauche avec le noyau  $\phi_{MQ}$  ( $\sigma = 4$ ,  $\mu = 1$ ) et en bas à droite avec le noyau  $\phi_{TPS}$  ( $\sigma = 2$ ).

type plaque mince, à influence globale, est utilisé pour l'étude de surfaces telles que les feuilles de papiers, les vêtements, les tissus (Bartoli et al., 2007; Bartoli and Zisserman, 2004; Gay-Bellile et al., 2007a, 2006; Vedaldi and Soatto, 2006) où sa propriété de déformation intrinsèquement lisse est bien adaptée. Il est aussi employé en imagerie médicale pour le recalage d'images du cerveau (Johnson and Christensen, 2002). Les fonctions plaque mince sont utilisées au chapitre 3 pour le recalage d'images d'une surface déformable.

### 2.3.1.4 Paramétrisation des fonctions à base radiale par primitives

Nous avons vu précédemment que les paramètres  $\mathbf{u}$  d'une fonction à base radiale peuvent être estimés linéairement à partir de correspondances de points choisis comme centre de la transformation et en imposant des conditions aux bords. Nous posons :

$$\begin{pmatrix} \mathbf{u}_x & \mathbf{u}_y \end{pmatrix} \stackrel{\text{def}}{=} \mathcal{E}_\lambda C', \quad (2.23)$$

avec

$$\mathcal{E}_\lambda = \begin{pmatrix} \mathcal{K}^{-1} \left( \mathbf{I} - \mathcal{P} (\mathcal{P}^\top \mathcal{K}^{-1} \mathcal{P})^{-1} \mathcal{P}^\top \mathcal{K}^{-1} \right) \\ (\mathcal{P}^\top \mathcal{K}^{-1} \mathcal{P})^{-1} \mathcal{P}^\top \mathcal{K}^{-1}, \end{pmatrix} \quad (2.24)$$

obtenu en inversant la matrice de transfert et  $C'^\top = \begin{pmatrix} \mathbf{c}'_1 & \cdots & \mathbf{c}'_p \end{pmatrix}$  la matrice rassemblant les centres sur l'image cible.

En combinant les équations (2.19) et (2.23) on obtient :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \mathbf{o}_q^\top \mathcal{E}_\lambda C', \quad (2.25)$$

avec  $\mathbf{u} = \text{vect}(C')$ . Ceci constitue la *paramétrisation par primitives* des fonctions à base radiale. Elles sont guidées par la position des centres  $\{\mathbf{c}'_j\}_{j=1}^p$  sur l'image cible.

Dans ce contexte, la fonction de déformation peut être vue comme un interpolant entre les primitives. La mise en correspondance de ces primitives entre deux images revient à définir une fonction de déformation. Cette dernière pouvant être utilisée pour transférer les primitives d'une image à l'autre, ou inversement pouvant être directement estimée à partir de ces primitives. Guider une déformation par des primitives a deux avantages principaux. Tout d'abord, il est plus intuitif de travailler sur la position de primitives, exprimés en pixels, plutôt que sur des coefficients difficilement interprétables. Ensuite, cette paramétrisation permet d'approximer les opérations d'inversion et de composition de fonctions, comme nous le verrons en §3.2. Celles-ci n'existent pour des transformations ne formant pas un groupe.

La figure 2.10 illustre le principe de déformation guidée par primitives. Une grille de  $3 \times 3$  centres est définie sur la région d'intérêt. Ses quatre coins sont étirés avec une amplitude de 2 pixels le long de leurs diagonales. La région d'intérêt est alors déformée en utilisant l'équation (2.25). La déformation obtenue est une interpolation de celle des centres, le type d'interpolation est directement lié au choix du noyau  $\phi$ .

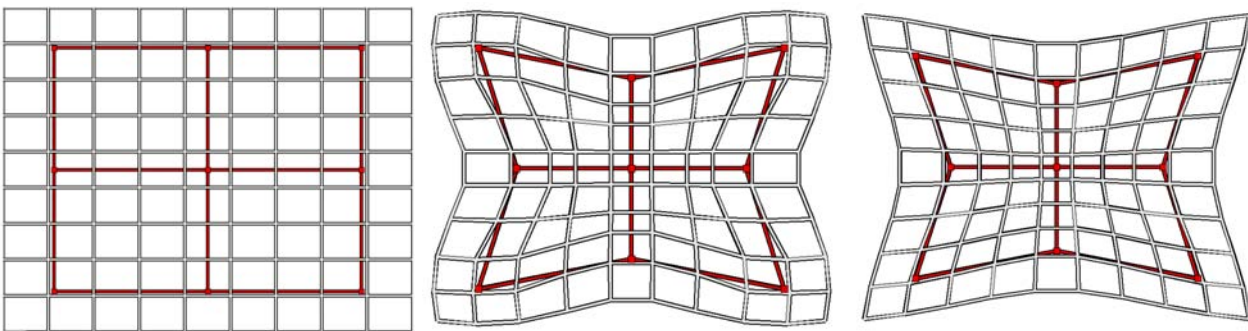


FIG. 2.10 – Exemples de fonctions à base radiale guidées par primitives. En rouge (ou gris foncé sur une version imprimée sans couleur) la grille des primitives, en blanc la grille représentant la région d'intérêt. A gauche : les positions de référence. Au milieu : déformation de la région d'intérêt par interpolation de type inverse multiquadratique ( $\mu = 3, \sigma = 4$ ). A droite : déformation de la région d'intérêt par interpolation de type plaque mince ( $\sigma = 1$ ).

## 2.3.2 Déformations de forme libre

### 2.3.2.1 Définition

Les *déformations de forme libre* (FFDs<sup>10</sup>) sont introduites dans (Sederberg and Parry, 1986) pour la déformation d'objets solides 3D en infographie. Elles sont également utilisées en recalage d'images notamment dans (Rueckert et al., 1999). Nous nous intéressons particulièrement aux FFDs de  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ . Le principe des FFDs est de déformer une région d'intérêt  $\mathcal{R}$  en manipulant un maillage régulier de points de contrôle. Notons que seule la région de l'espace située sous la grille de contrôle est déformée. Considérons une grille régulière de taille  $(p^x \times p^y)$ , dont l'espace-entres les points de contrôle  $\{\mathbf{c}_{i,j}\}_{i,j=1}^{p^x,p^y}$  suivant les axes  $x$  et  $y$  sont notés  $\delta^x$  et  $\delta^y$ . Le nombre

<sup>10</sup>"Free Form Deformations" en anglais



de points de contrôle est noté  $p : p = p^x p^y$ . Cette grille est superposée à une région d'intérêt  $\mathcal{R} = \{(x, y) : O^x \leq x \leq F^x, O^y \leq y \leq F^y\}$ , avec respectivement  $O^x$  et  $F^x$  les bornes inférieure et supérieure de la région d'intérêt le long de l'axe des  $x$  et  $O^y$  et  $F^y$  les bornes inférieure et supérieure de la région d'intérêt le long de l'axe des  $y$ . Soit  $\{\mathbf{c}'_{i,j}\}_{i,j=1}^{p^x, p^y}$  la grille de points de contrôle déformée, obtenue en perturbant la grille de référence :  $\mathbf{c}'_{i,j} \leftarrow \mathbf{c}_{i,j} + \delta \mathbf{c}_{i,j}$ . L'équation définissant une transformation par FFDs s'écrit :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \sum_{k=0}^d \sum_{l=0}^d B_k(u) B_l(v) \mathbf{c}'_{i+k, j+l}, \quad (2.26)$$

avec  $\mathbf{u}$  le vecteur de paramètres concaténant les  $\{\mathbf{c}'_{i,j}\}_{i,j=1}^{p^x, p^y}$  ;  $u, v$  sont les coordonnées normalisées de  $\mathbf{q}$  ;  $\{B_k\}_{k=1}^d, \{B_l\}_{l=1}^d$  sont les coefficients d'interpolation et  $d$  le degré de la fonction d'interpolation. En recalage d'images, la plus fréquemment utilisée est l'interpolation par B-spline. Les FFDs sont utilisées au chapitre 4 pour le recalage d'images d'une surface auto-occultée.

L'équation (2.26) peut être réécrite de la manière suivante :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \mathbf{b}_q^T \mathbf{C}', \quad (2.27)$$

avec  $\mathbf{b}_q^T$  le vecteur de taille  $(1 \times p)$  rassemblant les poids  $B(u)B(v)$  associé à chaque centre  $\mathbf{c}_{i,j}$  et  $\mathbf{C}'$  la matrice de taille  $(p \times 2)$  rassemblant les centres  $\{\mathbf{c}'_{i,j}\}_{i,j=1}^{p^x, p^y}$ .

### 2.3.2.2 Interpolation par B-spline

**Présentation générale.** Les *B-splines* sont des courbes polynomiales par morceaux, construites à partir d'un ensemble de points de contrôle. Soit  $d$  le degré des polynômes utilisés et  $p + 1$  le nombre de points de contrôle. Le  $i$ ème polynôme s'écrit :  $\mathcal{L}_i(t) = B_0(t)\mathbf{c}_i + B_1(t)\mathbf{c}_{i+1} + \dots + B_d(t)\mathbf{c}_{i+d}$ , avec  $i \in [0, p - d]$  et  $t \in [0, 1]$ . Les  $B_i(t)$  forment la base de polynôme de degré  $d$ . Cette base doit respecter certaines conditions : les courbes doivent se rejoindre au point de raccordement tout en assurant la continuité de leurs dérivées jusqu'à l'ordre  $d - 1$ . De plus la somme des coefficients  $\sum_{i=0}^d B_i(t)$  est égale à 1  $\forall t \in [0, 1]$ . Ces conditions permettent d'estimer les polynômes  $B_i(t)$ . Pour  $d = 3$ , on obtient après calcul la base de polynômes suivante :

$$\begin{aligned} B_0(t) &= \frac{1}{6}(-t^3 + 3t^2 - 3t + 1) & B_1(t) &= \frac{1}{6}(3t^3 - 6t^2 + 4) \\ B_2(t) &= \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) & B_3(t) &= \frac{1}{6}t^3. \end{aligned} \quad (2.28)$$

Elle est représentée sur la figure 2.11. Les principales propriétés des B-splines sont :

- ▷ **Un support compact.** Seuls  $d + 1$  points de contrôle influencent la position d'un point  $\mathbf{q} \in \mathcal{R}$ . En d'autres termes, déplacer un point de contrôle ne modifie que localement l'allure de la courbe.
- ▷ **La continuité des dérivées.** Elles sont assurées jusqu'à l'ordre  $d - 1$ .
- ▷ **La courbe ne passe pas par les points de contrôle.**
- ▷ **Région d'intérêt.** La courbe est située à l'intérieur de l'enveloppe convexe des points de contrôle.

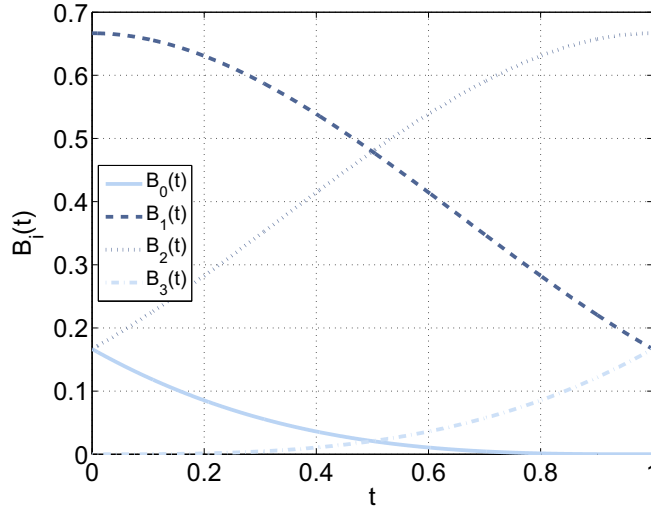


FIG. 2.11 – Base de polynômes pour des B-splines de degré 3.

**Application aux FFDs.** Dans le cadre des déformations de forme libre  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ , l'interpolation par B-spline entre les points de contrôle se fait suivant l'axe des  $x$  et l'axe des  $y$ . La déformation de la grille de contrôle est définie par un produit tensoriel de B-spline. Les coordonnées normalisées de  $\mathbf{q}$  sont données par :  $u = \frac{q^x - O^x}{\delta^x} - \left\lfloor \frac{q^x - O^x}{\delta^x} \right\rfloor$  et  $v = \frac{q^y - O^y}{\delta^y} - \left\lfloor \frac{q^y - O^y}{\delta^y} \right\rfloor$ , et les indices  $i$  et  $j$  par :  $i = \left\lfloor \frac{q^x - O^x}{\delta^x} \right\rfloor - 1$  et  $j = \left\lfloor \frac{q^y - O^y}{\delta^y} \right\rfloor - 1$ .

### 2.3.2.3 Illustration

L'exemple de la figure 2.9(a) est repris sur la figure 2.12. L'ensemble des noeuds constituant la grille est utilisé comme points de contrôle. La région d'intérêt est déformée en utilisant une simple interpolation bilinéaire et une interpolation par B-spline. Pour cette dernière, la déformation engendrée est locale : seule la région où les points de contrôle sont déplacés ainsi que son voisinage proche sont déformés. Elle est de plus naturellement lisse. Pour l'interpolation bilinéaire, la déformation introduite n'est pas lisse et des discontinuités apparaissent clairement au niveau de la région d'intérêt déformée.

### 2.3.2.4 Paramétrisation des déformations de forme libre par primitives

L'équation (2.27) ne définit pas directement la paramétrisation par primitives des FFDs. En effet les points de contrôle dans  $C$  (ou  $C'$ ) n'appartiennent pas à la surface engendrée par les FFDs. Une paramétrisation par primitives possible des FFDs est obtenue en projetant les points de contrôle sur la surface, engendrant le jeu de primitives  $C_s$  :

$$C_s = T_c C \quad \text{avec} \quad T_c = \begin{pmatrix} \mathbf{b}_{c_1}^T \\ \vdots \\ \mathbf{b}_{c_p}^T \end{pmatrix}.$$

Notons que la matrice  $T_c$  est de taille  $(p \times p)$ . La paramétrisation par primitives des FFDs est

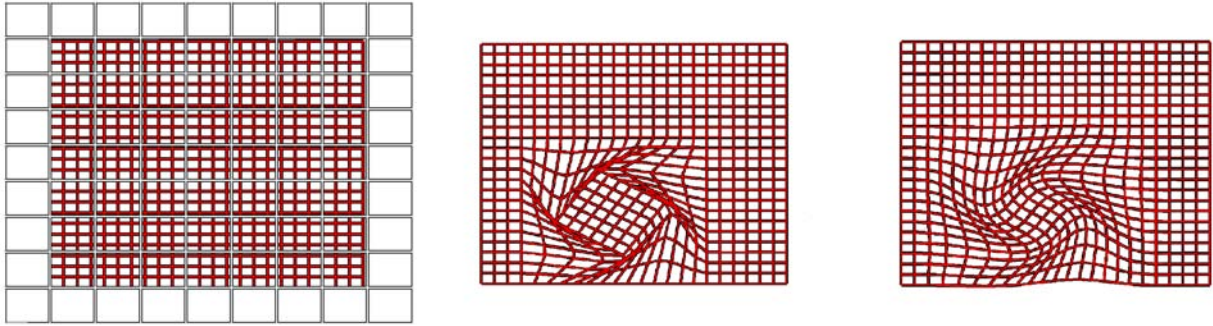


FIG. 2.12 – **Exemples de déformations par FFDs.** L'exemple synthétique généré est présenté sur la figure 2.9(a). A gauche : en blanc la grille des points de contrôle, en rouge (ou gris foncé sur une version imprimée sans couleur) la région d'intérêt. Au centre : déformation de la région d'intérêt avec une interpolation bilinéaire. A droite : déformation de la région d'intérêt avec une interpolation par B-spline.

finaleme nt donnée par :

$$\mathcal{W}(\mathbf{q}; \mathbf{u}) = \mathbf{b}_q^T \mathbb{T}_c^{-1} \mathbf{C}'_s, \quad (2.29)$$

avec  $\mathbf{C}'_s \leftarrow \mathbf{C}_s + \delta \mathbf{C}_s$ . La matrice  $\mathbb{T}_c$  est constante puisqu'elle ne dépend que des centres sur la grille régulière de référence. Son « inverse » peut donc être pré calculée.

## 2.4 Modélisation des déformations 3D par une combinaison linéaire de modes

Dans la section précédente nous avons présenté différentes fonctions de déformation modélisant les déformations image. Cette section est consacrée à la modélisation 3D des déformations d'une surface. Plus précisément nous présentons les différents modèles approxima nt les déformations 3D d'une surface par une combinaison linéaire de *modes de déformation*.

### 2.4.1 Définitions des modèles de faible rang, « morphables » et actifs

Les *modèles de faible rang* (LRSMs<sup>11</sup>), les *modèles morphables* (3DMMs<sup>12</sup>) et les *modèles actifs* (AMs<sup>13</sup>) ont en commun de représenter les déformations d'une surface  $\{\mathbf{S}_j\}_{j=1}^m$  ( $m$  le nombre de point utilisé pour représenter cette dernière), par une combinaison linéaire pondérée de modes de déformation. De manière générique ils sont décrits par l'équation suivante :

$$\mathbf{S}_j \stackrel{\text{def}}{=} \sum_{k=1}^l a_k \mathbf{B}_{k,j}, \quad (2.30)$$

où  $l$  est le nombre de modes de déformation, les  $\{\mathbf{B}_{k,j}\}_{k,j=1}^{l,m}$  sont les modes de déformation et les  $\{a_k\}_{k=1}^l$  sont les *coefficients de forme*.

<sup>11</sup>"Low Rank Shape models" en anglais.

<sup>12</sup>"3D Morphable Models" en anglais.

<sup>13</sup>"Active Models" en anglais.

Les différences entre ces modèles proviennent de deux caractéristiques majeures :

- ▷ **Dimension 2D ou 3D.** Un modèle 3D combine un modèle de caméra avec une forme 3D déformable. Son avantage est qu'il contient directement la pose de la caméra et la structure 3D. Il est en revanche plus « non-linéaire » qu'un modèle 2D.
- ▷ **Apprentissage : pré-appris ou non appris.** Un *modèle pré-appris* est dédié à un type de surface : les modes de déformation sont estimés sur une base d'apprentissage ou par étude physique des déformations de la surface. Seuls les coefficients de forme et le mouvement global doivent être estimés sur les données courantes (Blanz and Vetter, 1999; Salzmann et al., 2005). Il a été récemment proposé d'estimer les modes de déformation directement sur les données courantes (Bregler et al., 2000; Boulton and Brown, 1991; Irani, 1999) : on parle alors de *modèle non appris*. Les modèles pré-appris sont plus stables et mieux posés, mais moins génériques.

Le tableau 2.5 résume les différentes terminologies employées en fonction des propriétés du modèle. Un algorithme de reconstruction 3D basé sur le modèle de faible rang est proposé au chapitre 5.

	2D	3D
Pré-appris	AM (modèle actif) linéaire	3DMM (modèle morphable 3D) au moins bilinéaire
Non appris	LRSM (modèle de faible rang) <i>implicite</i> bilinéaire	LRSM (modèle de faible rang) <i>explicite</i> au moins trilineaire

TAB. 2.5 – Les différents modèles représentant les déformations d'une surface par une combinaison linéaire de modes de déformation. Nous indiquons pour chacun d'entre eux leur complexité par rapport aux paramètres inconnus. Ce tableau récapitulatif est extrait de (Bartoli, 2008).

## 2.4.2 Quelques exemples de modèles pré-appris

**Modèles actifs de forme et d'apparence.** Ces modèles sont proposés dans (Cootes and Taylor, 2001). Le *modèle actif de forme* (ASM<sup>14</sup>) représente les variations de forme d'une classe d'objet tandis que le *modèle actif d'apparence* (AAM<sup>15</sup>) encode les variations de forme et d'apparence. L'AAM est principalement utilisé pour la modélisation de visages (Cootes et al., 1998; Matthews and Baker, 2004) mais peut néanmoins être appliqué à une grande variété d'objets (Sclaroff and Isidoro, 1998). Le modèle AAM de visage proposé par Cootes *et al.* est composé de 68 sommets. Les variations de forme se traduisent par la position des sommets du modèle, celles d'apparence par la valeur des pixels. L'AAM et l'ASM sont construits par ACP<sup>16</sup> d'une base d'apprentissage annotée, la plupart du temps manuellement. La figure 2.13 illustre les modes de déformation (encodant la forme) obtenus par ACP d'une base de visage annotée. Le modèle AAM de visage est utilisé en §5.5.2 pour suivre les expressions d'un visage sur une vidéo. Les points suivis sont alors fournis à l'entrée de l'algorithme de reconstruction 3D proposé en §5.4.

### Modèles morphables 3D.

<sup>14</sup>"Active Shape Model" en anglais.

<sup>15</sup>"Active Appearance Model" en anglais.

<sup>16</sup>Analyse en Composantes Principales.

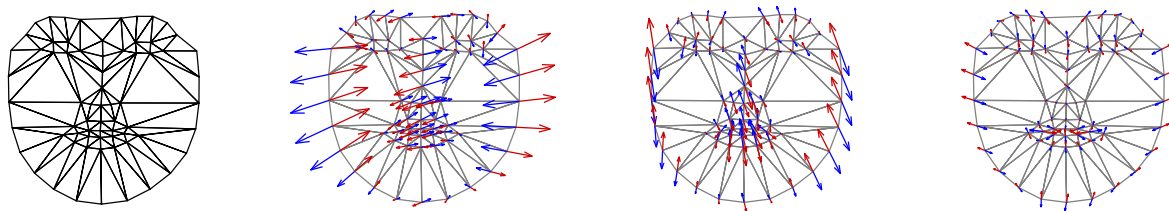


FIG. 2.13 – Exemples de modes de déformation pour le modèle actif de visage de Cootes *et al.* (images extraites de (Matthews and Baker, 2004)).

**Modèle morphable de visage.** Ce modèle est proposé dans (Blanz and Vetter, 1999). Leur base de données se compose d'environ 400 visages scannés par un Laser. Les visages résultants sont composés de 70.000 sommets 3D plus l'information couleur associée. Un modèle linéaire de forme et un modèle linéaire d'apparence sont appris par ACP de la base d'apprentissage. La mise en correspondance des données d'apprentissage se fait automatiquement par une approche de type *flot optique*. Néanmoins, en raison des grandes variétés d'apparence d'un visage mais également en raison de la présence de régions à faible texture, certaines correspondances sont erronées. Le modèle appris présente alors quelques ambiguïtés.

L'expérimentation qui suit illustre ce problème. Un sous ensemble de sommets est sélectionné sur un visage de référence. Un nouveau visage est ensuite généré en perturbant les paramètres de forme et d'apparence. Les sommets sélectionnés sont ensuite affichés sur le visage obtenu. La figure 2.14 montre clairement certaines ambiguïtés du modèle. En effet, les sommets correspondant aux sourcils sur le visage de référence se situent juste au-dessus des yeux sur le visage généré.

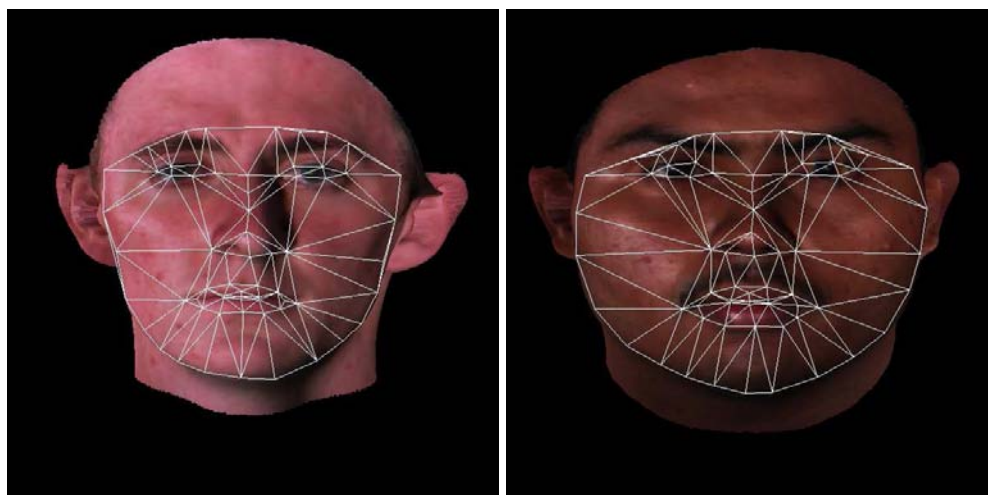


FIG. 2.14 – Illustration des ambiguïtés du modèle morphable 3D proposé par Blanz et Vetter. Ces expérimentations ont été réalisées par Hugo Mercier.

**Modèle morphable de surfaces planes.** Dans (Salzmann et al., 2005) un modèle morphable 3D approximant les déformations de surfaces telles qu'une feuille de papier, un tissu, *etc.* est proposé. La

surface est représentée par un maillage 3D triangulaire. Les exemples d'apprentissages sont obtenus en perturbant synthétiquement les angles entre les facettes du maillage défini comme inextensible. Ils montrent que 4 familles d'angles suffisent pour définir l'ensemble des déformations possibles pour ce type de surface. Ces angles sont représentés sur la figure 2.15.

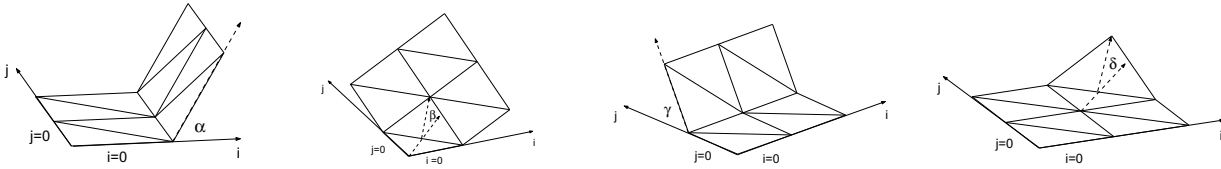


FIG. 2.15 – Les quatre familles d'angles :  $\alpha, \beta, \gamma$  et  $\delta$  permettant de construire le modèle morphable de surfaces planaires proposé par Salzmann *et al.* (images extraites de (Salzmann *et al.*, 2005))

La famille d'angles  $\alpha$  code l'orientation des triangles inférieurs de la première ligne du maillage, la famille d'angles  $\beta$  modifie l'orientation des triangles supérieurs de la première ligne du maillage. La famille d'angles  $\gamma$  oriente les triangles inférieurs de la première colonne du maillage. Enfin, la famille d'angles  $\delta$  code l'orientation des triangles supérieurs de la dernière colonne du maillage. Les autres sommets du maillage sont contraints par la propriété d'inextensibilité. Les figures 2.16(a), 2.16(b), 2.16(c), 2.16(d) illustrent des exemples d'apprentissages obtenus en perturbant respectivement les familles d'angles  $\alpha, \beta, \delta$  et  $\gamma$ . Notons que le modèle génératif de papier proposé dans (Perriollat and Bartoli, 2007) permet de générer des exemples d'apprentissage semblables.

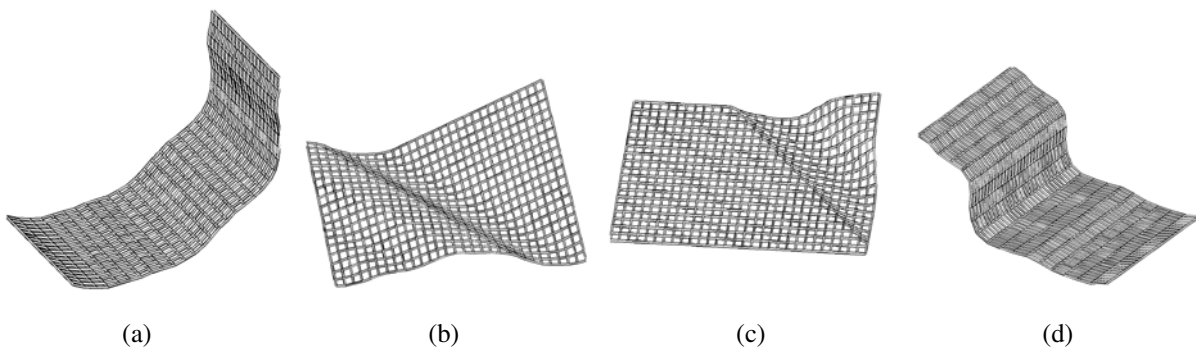


FIG. 2.16 – **Illustration des données d'apprentissage du modèle morphable de surfaces planaires proposé par Salzmann *et al.*** (a) Exemple de déformation obtenue en faisant varier l'angle  $\alpha$ . (b) Exemple de déformation obtenue en faisant varier l'angle  $\beta$ . (c) Exemple de déformation obtenue en faisant varier l'angle  $\delta$ . (d) Exemple de déformation obtenue en faisant varier l'angle  $\gamma$ .

L'apprentissage se fait par ACP. Les figures 2.17(a), 2.17(b), 2.17(c), 2.17(d) illustrent certains modes de déformation du modèle morphable de Salzmann *et al.* Des modes (par exemple le mode 2) codent la dilatation ou la contraction du maillage suivant les axes  $x$  et  $y$  alors que les exemples d'apprentissage présentent tous les mêmes dimensions.

L'inextensibilité est une contrainte non-linéaire qui n'est pas préservée lors de l'ACP. Ceci peut être vu à la fois comme un avantage et comme un inconvénient. Le modèle obtenu représente un spectre de déformations plus large mais il est par conséquent moins contraint.

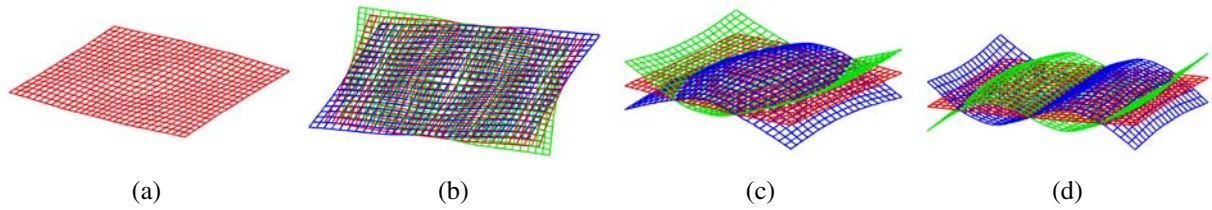


FIG. 2.17 – **Exemples de déformations introduites par le modèle morphable de surfaces planes proposé par Salzmann *et al.*** (a) La forme moyenne. (b) Déformation introduite par le mode 1. (c) Déformation introduite par le mode 7. (d) Déformation introduite par le mode 12.

**Le modèle *Candide-3*.** Le modèle *Candide-3* est proposé dans (Ahlberg, 2001). Il représente la forme d'un visage par un maillage 3D composé de 113 sommets. Ce modèle varie selon des paramètres morphologiques et d'expressions ne dérivant pas d'une analyse statistique mais issus d'une étude physique. Il découle des normes MPEG<sup>17</sup>-4 pour l'animation de visage. La formule générale décrivant le modèle *Candide-3* est donnée par :

$$\mathbf{S}_j \stackrel{\text{def}}{=} \sum_{k=1}^{l^f} a_k^f \mathbf{B}_{k,j}^f + \sum_{k=1}^{l^e} a_k^e \mathbf{B}_{k,j}^e. \quad (2.31)$$

Les  $\{\mathbf{B}_{k,j}^f\}_{k,j=1}^{l^f,m}$ ,  $\{\mathbf{B}_{k,j}^e\}_{k,j=1}^{l^e,m}$  représentent respectivement les *unités de formes* (SUs<sup>18</sup>) et les *unités d'actions* (AUs<sup>19</sup>) faciales ou d'expression ; les coefficients  $\{a_k^f\}_{k=1}^{l^f}$  et  $\{a_k^e\}_{k=1}^{l^e}$  représentent respectivement les paramètres de la forme du visage et les paramètres d'expression.

Les SUs permettent d'adapter le modèle 3D à la physionomie d'une personne. Ces dernières appliquent un déplacement sur un ensemble réduit de points qui régissent la forme du nez, la taille de la bouche, la largeur du menton, *etc.* L'ensemble des SUs est répertorié dans le tableau 2.6 ; des exemples sont illustrés sur la figure 2.18.

Les AUs codent les modifications physiques d'un visage issues de l'activation de muscles faciaux. Elles permettent de reproduire les expressions d'un visage. Ces dernières appliquent un déplacement sur un ensemble réduit de points qui régissent l'ouverture de la bouche, le froncement des sourcils, l'abaissement des paupières *etc.* L'ensemble des AUs est répertorié dans le tableau 2.7 et des exemples sont donnés sur la figure 2.19.

Ce modèle est utilisé pour des applications de suivi d'expressions faciales (Chen and Davoine, 2006; Dornaika and Davoine, 2005) et de reconstruction 3D (Chaumont and Beaumesnil, 2005). Nous l'utilisons pour générer une vidéo de synthèse représentant un visage animée. Les données générées nous permettent de quantifier la précision des déformations reconstruites par l'algorithme de reconstruction 3D de surfaces déformables que nous proposons en §5.4.

<sup>17</sup>"Moving Picture Experts Group" en anglais.

<sup>18</sup>"Shape Units" en anglais.

<sup>19</sup>"Action Units" en anglais.

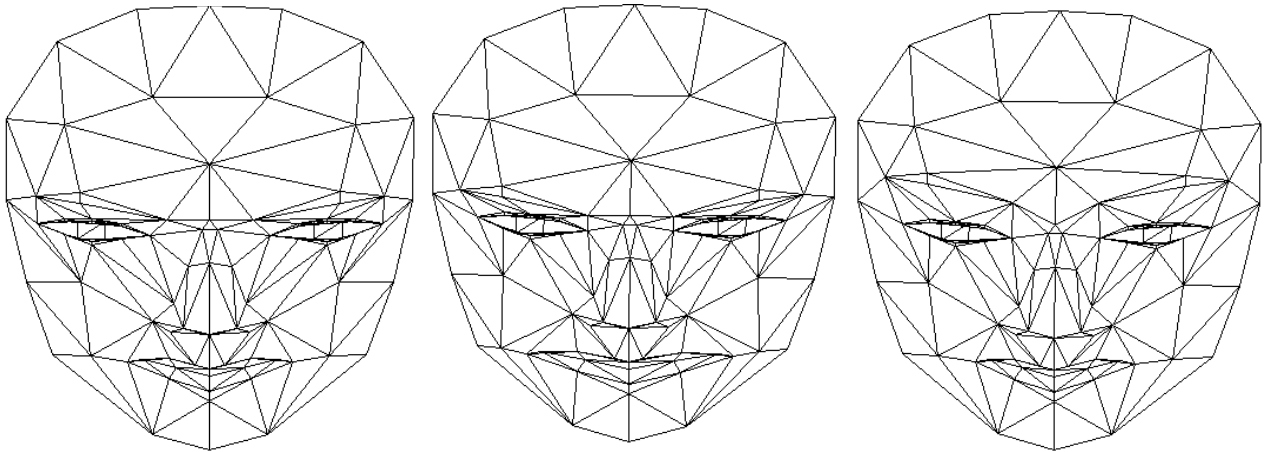


FIG. 2.18 – Exemples d’unités de forme du modèle *Candide-3*. A gauche : les yeux sont écartés (SU n°6). Au centre : la bouche est étirée (SU n°12). A droite : les sourcils sont relevés (SU n°2).

Indice	AU
1	Lever la lèvre supérieure
2	Abaissier la lèvre inférieure
3	Étirer horizontalement les lèvres
4	Froncer les sourcils
5	Étirer verticalement les lèvres
6	Lever les sourcils
7	Fermeture des yeux
8	Mouvement des paupières
9	Mouvement du nez
10	Pincer les lèvres
11	Relever les paupières

TAB. 2.6 – Les AUs du modèle *Candide-3*

Indice	SU
1	Largeur du front
2	Position verticale des sourcils
3	Position verticale des yeux
4	Longueur des yeux
5	Largeur des yeux
6	Séparation horizontale des yeux
7	Gonflement des joues
8	Forme du nez
9	Position verticale du nez
10	Forme du bout du nez
11	Position verticale de la bouche
12	Largeur de la bouche
13	Décalage vertical des yeux
14	Largeur du menton

TAB. 2.7 – Les SUs du modèle *Candide-3*

## 2.5 Conclusion

Nous avons présenté les différents modèles de déformation (basé image et 3D) utilisés dans notre étude ainsi que les méthodes permettant de les estimer à partir d’informations extraites des images. Elles se classent en deux groupes, les méthodes basées primitives et celles dites directes. Ces dernières sont privilégiées dans nos travaux pour leur grande précision. Nous allons voir dans le chapitre suivant comment étendre les algorithmes compositionnels aux modèles déformables. Ils ne peuvent théoriquement pas être utilisés pour estimer une fonction de déformation de par leur contrainte de groupe. Nous proposons d’approximer les opérations d’inversion et de composition de fonctions par divers processus basés sur la paramétrisation par primitives des fonctions de déformation.



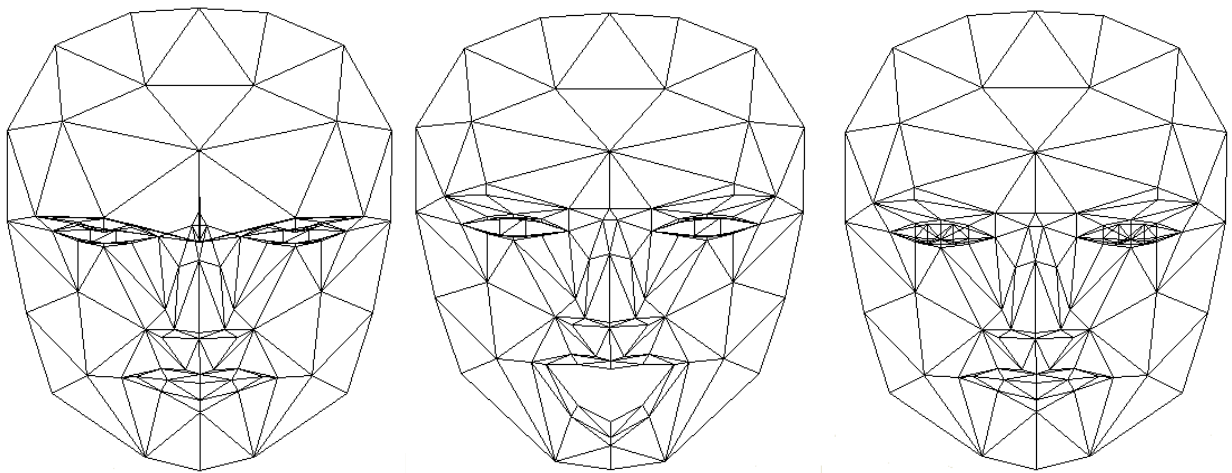


FIG. 2.19 – Exemples d’unités d’action du modèle *Candide-3*. A gauche : le froncement des sourcils (AU n°4). Au centre : l’abaissement de la mâchoire inférieure (AU n°2). A droite : la fermeture des yeux (AU n°7).



# Chapitre 3

## Algorithmes compositionnels pour modèles déformables

---

---

Dans ce chapitre, nous proposons d'étendre les algorithmes de recalage d'images à mise à jour compositionnelle aux modèles déformables. Les algorithmes à mise à jour additive sont habituellement privilégiés pour estimer une fonction de déformation puisque les étapes de composition et d'inversion n'existent pas pour des transformations ne formant pas un groupe. Nous apportons deux contributions. Tout d'abord, le principe du guidage des déformations par des primitives géométriques est proposé. Il permet l'emploi d'algorithmes compositionnels pour certaines fonctions de déformation. Ensuite, une approche par apprentissage, basée sur un modèle linéaire par morceaux de la fonction de coût, est utilisée pour l'étape de recalage local. La combinaison de ces deux contributions engendre un algorithme dont le bassin de convergence est élargi, les temps de calcul diminués et la précision améliorée, en comparaison des algorithmes « classiques » à mise à jour additive. Nous avons appliqué ces travaux à l'augmentation 2D d'images d'une surface déformable et à la capture de déformations. Ils ont été publiés dans (Gay-Bellile et al., 2007a,e).

---

---

### 3.1 Introduction

L'objectif du chapitre est de mettre en place un algorithme de recalage d'images d'une surface déformable (feuille de papier, tissu, *etc.*) qui soit à la fois précis et efficace en terme de temps de calcul. Les applications visées sont principalement la *réalité augmentée*, plus précisément l'augmentation 2D de surfaces déformables sur une vidéo, et la capture de déformations. Le champ de déplacement dense obtenu peut également servir comme donnée d'entrée d'un algorithme de reconstruction 3D basé sur le modèle de faible rang ; comme par exemple celui proposé en §5.4.

Un algorithme de recalage performant est proposé dans (Pilet et al., 2008). Un critère basé primitives est minimisé, ce qui engendre quelques difficultés : il est indispensable de lever l'ambiguïté entre les déformations réelles de la surface et les erreurs de mises en correspondance. Pour palier à cette difficulté, les auteurs proposent le schéma d'optimisation suivant : pour les premières itérations, le terme de régularisation est prédominant par rapport au terme de données (transformation quasi rigide) puis l'influence de la régularisation est réduite au fur et à mesure que les erreurs de mises en

correspondance sont supprimées par un estimateur robuste ; les déformations du modèle sont alors encouragées. Leur procédé donne de très bon résultats mais des imprécisions peuvent néanmoins apparaître, l'ambiguïté étant dans certains cas difficile à lever. Le résultat du recalage doit alors être raffiné par la minimisation d'un critère direct.

Nous nous concentrons dans ce chapitre sur les méthodes directes de recalage. La difficulté décrite ci-dessus est alors évitée. Ces méthodes sont plus précises de par la prise en compte d'un grand nombre de données, mais sont en contrepartie coûteuses en temps de calcul. En effet, la fonction de coût est habituellement minimisée par l'algorithme DA-GN<sup>1</sup>, pour lequel la matrice Hessienne doit être estimée et inversée à chaque itération. Notons qu'un schéma à complexité variable de la fonction de déformation peut être utilisé : insertion de centres ou points de contrôle (Bartoli and Zisserman, 2004), réglage du coefficient de régularisation au cours des itérations (Lim and Yang, 2005). Les algorithmes compositionnels, par exemple l'algorithme IC-GN<sup>2</sup>, sont plus efficaces mais ne peuvent être utilisés sous leur forme d'origine avec des modèles déformables. Ils requièrent une transformation formant un groupe ce qui n'est pas le cas de la plupart des fonctions de déformation. L'extension de ces algorithmes aux transformations ne formant pas de groupe a été étudiée pour des modèles morphables (Romdhani and Vetter, 2003) et actifs (Matthews and Baker, 2004). Les étapes d'inversion et de composition sont approximées. La composition est par exemple réalisée en opérant directement sur la position des sommets du maillage (ou de leur projection pour des modèles morphables). Certaines contraintes relatives aux modèles ne sont alors pas prises en compte. Une étape supplémentaire est nécessaire afin d'imposer ces contraintes.

Une fonction de déformation peut également être évaluée efficacement en apprenant hors ligne la relation entre la variation d'intensité lumineuse (ou de couleur) et la variation locale des paramètres. Cette relation n'est valide que localement, c'est-à-dire autour des paramètres de référence, nécessitant une loi de mise à jour compositionnelle. L'apprentissage permet un gain substantiel en terme de temps de calcul par rapport aux algorithmes tels que DA-GN et DA-ESM<sup>3</sup>, en effet, l'estimation et l'inversion de la matrice Hessienne sont remplacées par une simple application de la relation apprise.

Nous proposons deux contributions principales. La première, présentée en §3.2, est le concept de recalage d'images guidé par primitives. Il permet l'emploi d'algorithmes compositionnels en relaxant la contrainte de groupe pour la plupart des fonctions de déformation comme par exemple les RBFs<sup>4</sup> (avec les noyaux décrits en §2.3.1.3 : plaque mince, multiquadrique, inverse multiquadrique, etc.). L'idée consiste à guider la fonction de déformation par un ensemble de primitives, et d'opérer directement sur ces dernières pour réaliser des opérations telles que le *renversement* et l'*enchaînement* de déformations, approximant respectivement l'inversion et la composition de fonctions. Celles-ci n'existent pas pour des transformations ne formant pas un groupe. Nous montrons par exemple que l'algorithme IC-GN et ceux basés apprentissage peuvent être utilisés avec des fonctions de déformation de type plaque mince.

La deuxième contribution, présentée en §3.3, est une amélioration des algorithmes basés apprentissage. Les travaux précédents (Cootes et al., 1998; Jurie and Dhome, 2002a) approximent linéairement la relation entre la variation d'intensité et la variation locale des paramètres. Ceci implique plusieurs problèmes en pratique, comme par exemple une faible précision du recalage. Afin de palier à ces inconvénients, une approximation linéaire par morceaux est proposée. D'autres approximations

<sup>1</sup>DA-GN pour "Direct Additive Gauss-Newton", voir §2.2.2.3.

<sup>2</sup>IC-GN pour "Inverse Compositional Gauss-Newton", voir §2.2.2.3.

<sup>3</sup>DA-ESM pour "Direct Additive Efficient Second order Minimisation", voir §2.2.2.3.

<sup>4</sup>"Radial Basis Functions".

(Agarwal and Triggs, 2006; Chateau et al., 2008) existent ; elles ne sont pas étudiées dans ce chapitre.

La combinaison du concept de recalage d'images guidé par primitives avec l'approche basée apprentissage proposée pour le recalage local permet de surpasser les autres algorithmes sur la plupart des expérimentations. Il est montré expérimentalement que les performances en terme de précision de recalage, temps de calcul et largeur du bassin de convergence sont grandement améliorées.

**Organisation du chapitre.** Le concept de recalage d'images guidé par primitives est décrit en §3.2. Ensuite, le recalage d'images basé sur un apprentissage hors ligne de la relation intensité / déplacement, ainsi que l'amélioration proposée sont introduits en §3.3. Pour finir, nos résultats expérimentaux sont présentés en §3.4 et nous concluons et discutons ce chapitre en §3.5.

## 3.2 Recalage compositionnel guidé par primitives

### 3.2.1 Travaux antérieurs

Les approches existantes (Matthews and Baker, 2004; Romdhani and Vetter, 2003) qui ont préalablement étendu les algorithmes compositionnels, notamment IC-GN décrit en §2.2.2.3, aux transformations ne formant de groupe, utilisent un modèle morphable ou actif dont les sommets sont triangulés :

$$\underbrace{\mathbf{s}_j^r = \sum_{k=1}^l a_k \mathbf{B}_{k,j}}_{\text{modèle actif}} \quad \text{ou} \quad \underbrace{\mathbf{s}_j^r = \Pi \left( \sum_{k=1}^l a_k \mathbf{B}_{k,j} \right)}_{\text{modèle morphable}}.$$

La position des sommets dans l'image dépend d'un sous ensemble de paramètres  $\{a_k\}_{k=1}^l$  et est contrainte par les modes de déformation appris préalablement.

Une fonction de déformation de type affine par morceaux est définie entre les sommets (ou leurs projections pour les modèles morphables) du modèle, contenu dans  $\mathbf{s}$ . L'étape de composition est approximée en opérant directement sur ces sommets, plus de détails sont donnés en §3.2.1.1. La forme obtenue  $\mathbf{s} + \Delta\mathbf{s}$ , après mise à jour, ne correspond pas forcément à une instance du modèle puisque les contraintes relatives aux modes de déformation ne sont pas prises en compte lors de cette opération. Une étape supplémentaire est donc requise afin de forcer les contraintes et d'estimer les paramètres du modèle à partir de la forme mise à jour. Pour cela, l'erreur entre la forme  $\mathbf{s} + \Delta\mathbf{s}$  et celle induite par le modèle  $\mathbf{s}^r$ , est minimisée :

$$\min_{\{a_k\}_{k=1}^l, \mathbf{D}} \|\mathbf{s} + \Delta\mathbf{s} - \mathbf{D}\mathbf{s}^r\|^2, \quad (3.1)$$

avec  $\mathbf{s}^r \stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{s}_1^r & \cdots & \mathbf{s}_m^r \end{pmatrix}$  et  $\mathbf{D}$  le mouvement global.

#### 3.2.1.1 Approximation de la composition

Soient les paramètres courants  $\mathbf{u}$  pour lesquels sont associés les sommets  $\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 & \cdots & \mathbf{s}_m \end{pmatrix}$ , avec  $m$  le nombre de sommets constituant le modèle. De même pour les paramètres de référence  $\mathbf{u}_0$ , la forme associée est définie par  $\mathbf{s}_0 = \begin{pmatrix} \mathbf{s}_{0,1} & \cdots & \mathbf{s}_{0,m} \end{pmatrix}$ . On suppose également que la fonction de

déformation locale  $\mathcal{W}(\cdot; \Delta \mathbf{u})$  est estimée : la variation des paramètres  $\Delta \mathbf{u}$  engendre une perturbation de la forme de référence  $\mathbf{s}_0 + \Delta \mathbf{s}_0$  avec  $\Delta \mathbf{s}_0 = \begin{pmatrix} \Delta \mathbf{s}_{0,1} & \cdots & \Delta \mathbf{s}_{0,m} \end{pmatrix}$ .

Le processus de composition basé sur une mise à jour de la forme courante  $\mathbf{s}$  consiste à estimer  $\Delta \mathbf{s}$  à partir de  $\Delta \mathbf{s}_0$ . Considérons le sommet  $j$  pour lequel la mise à jour locale  $\Delta \mathbf{s}_{0,j}$  est connue. Ce sommet est lié à différents triangles pour lesquels une transformation affine est définie entre la forme de référence et la forme courante. Suivant le triangle sélectionné la transformation induite est différente, impliquant différentes mises à jour  $\Delta \mathbf{s}_j$ . Une possibilité est de choisir la transformation affine relative au triangle contenant  $\mathbf{s}_{0,j} + \Delta \mathbf{s}_{0,j}$ . Dans (Matthews and Baker, 2004), une alternative est proposée. Elle est illustrée sur la figure 3.1. Plusieurs mises à jour  $\Delta \mathbf{s}_{j,k}$  sont appliquées à  $\mathbf{s}_{0,j} + \Delta \mathbf{s}_{0,j}$  les différentes transformations affines associées aux  $k$  triangles dont  $\mathbf{s}_j$  est un sommet. Les mises à jour obtenues sont au final moyennées pour aboutir à  $\Delta \mathbf{s}_j$ . Ce procédé a pour effet de lisser la forme  $\mathbf{s} + \Delta \mathbf{s}$ .

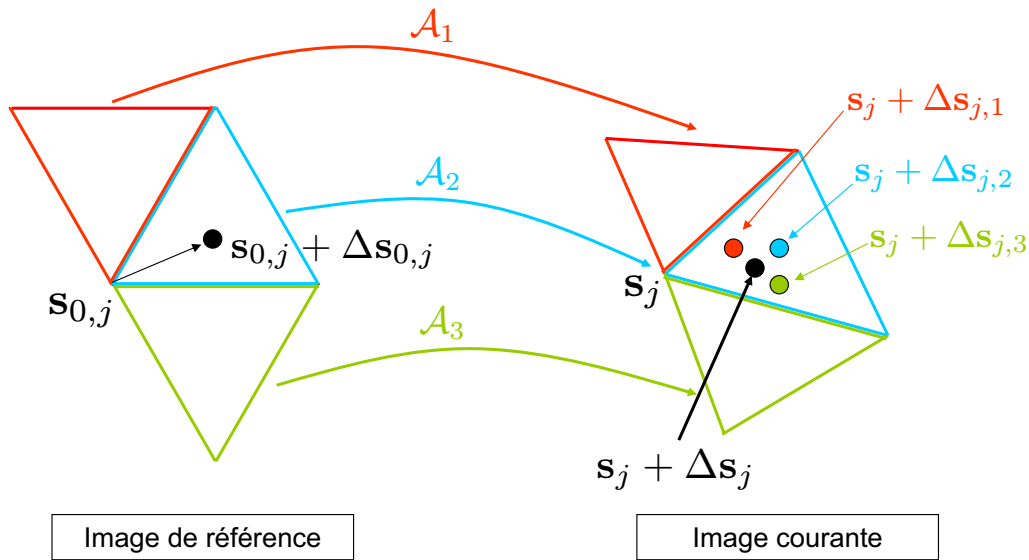


FIG. 3.1 – **Composition approximative de transformations affines par morceaux proposée dans (Matthews and Baker, 2004).** La mise à jour  $\Delta \mathbf{s}_j$  du sommet  $\mathbf{s}_j$  est obtenue en moyennant les différentes mises à jour engendrées par les triangles dont  $\mathbf{s}_j$  est un sommet.

### 3.2.1.2 Approximation de l'inversion

Pour l'algorithme IC-GN, la fonction de déformation locale  $\mathcal{W}(\cdot; \Delta \mathbf{u})$  doit être inversée avant la mise à jour compositionnelle. Pour cela, différentes approximations ont été proposées.

L'inversion d'une fonction de déformation est approximée dans (Matthews and Baker, 2004) par un développement de Taylor au premier ordre :

$$\mathcal{W}(\mathbf{q}; \Delta \mathbf{u}) = \mathbf{q} + \frac{\partial \mathcal{W}}{\partial \mathbf{u}}(\mathbf{q}; \mathbf{u}_0) \Delta \mathbf{u} + \varepsilon(\Delta \mathbf{u}^2).$$

Si l'on compose  $\mathcal{W}(\mathbf{q}; \Delta \mathbf{u})$  et  $\mathcal{W}(\mathbf{q}; -\Delta \mathbf{u})$ , on obtient :

$$\mathcal{W}(\mathbf{q}; \Delta \mathbf{u}) \circ \mathcal{W}(\mathbf{q}; -\Delta \mathbf{u}) = \mathbf{q} - \frac{\partial \mathcal{W}}{\partial \mathbf{u}}(\mathbf{q}; \mathbf{u}_0) \Delta \mathbf{u} + \frac{\partial \mathcal{W}}{\partial \mathbf{u}}(\mathbf{q}; \mathbf{u}_0) \Delta \mathbf{u} + \varepsilon(\Delta \mathbf{u}^2) = \mathbf{q} + \varepsilon(\Delta \mathbf{u}^2).$$

Le résultat de la composition de  $\mathcal{W}(\mathbf{q}; -\Delta\mathbf{u})$  et  $\mathcal{W}(\mathbf{q}; \Delta\mathbf{u})$  est donc égal à la transformation identité au premier ordre. Dans (Matthews and Baker, 2004) l'inverse de la fonction de déformation locale est donc obtenue par :

$$\mathcal{W}(\mathbf{q}; \Delta\mathbf{u})^{-1} \approx \mathcal{W}(\mathbf{q}; -\Delta\mathbf{u}).$$

Une alternative est proposée dans (Romdhani and Vetter, 2003), en se basant sur la structure triangulaire du maillage. La fonction de déformation locale  $\mathcal{W}(\cdot; \Delta\mathbf{u})$  est appliquée à chaque sommet du maillage, transformant un triangle  $\xi'$  en un triangle  $\xi$ . La fonction de déformation inverse évaluée en un point  $\mathbf{q}$  appartenant à  $\xi$  :  $\mathbf{q}' = \mathcal{W}(\mathbf{q}; \Delta\mathbf{u})^{-1}$  est obtenue en positionnant  $\mathbf{q}'$  tel que la position relative de  $\mathbf{q}'$  dans  $\xi'$  soit la même que celle de  $\mathbf{q}$  dans  $\xi$ . Cette opération garantit que  $\mathcal{W}(\mathcal{W}(\mathbf{q}; \Delta\mathbf{u})^{-1}; \Delta\mathbf{u}) = \mathbf{q}$ .

### 3.2.1.3 Discussion

Nous avons vu précédemment que les algorithmes compositionnels peuvent être étendus pour des déformations engendrées par un modèle morphable ou actif. Les processus présentés, approximant l'inversion et la composition de fonctions, s'appuient sur le maillage triangulé de ces modèles. Une étape supplémentaire est en outre requise afin de réinjecter les contraintes du modèle à l'issue du processus de composition.

Nous allons voir ci-dessous comment étendre les algorithmes compositionnels pour des fonctions de déformation comme par exemple les RBFs. Notre approche s'appuie sur le principe du guidage des déformations par des primitives géométriques. Nous proposons deux solutions empiriques basées sur la paramétrisation par primitives pour la composition et l'inversion, nommées respectivement enchaînement et renversement. Ces opérations prennent directement en compte les contraintes relatives à la fonction de déformation ; aucune étape supplémentaire n'est alors requise. Grâce à ces opérations l'algorithme IC-GN « classique » devient l'algorithme IC-GN guidé par primitives pour lequel les fonctions plaque mince peuvent être utilisées.

## 3.2.2 Le guidage par primitives

Le principe du guidage par primitives des fonctions de déformation est de paramétrer ces dernières par un ensemble de primitives géométriques. La fonction de déformation peut alors être vue comme un interpolant entre les primitives. La mise en correspondance de ces primitives entre deux images revient à définir une fonction de déformation. Cette dernière pouvant être utilisée pour transférer les primitives d'une image à l'autre, ou inversement pouvant être directement estimée à partir de ces primitives. Les primitives ont une position fixe sur l'image de référence qui dépend de la fonction de déformation utilisée.

Guider une déformation par des primitives a deux avantages principaux. Tout d'abord, il est plus intuitif de travailler sur la position de primitives, exprimés en pixels, plutôt que sur des coefficients difficilement interprétables. Ensuite, cette paramétrisation permet d'approximer les opérations d'inversion et de composition de fonctions, comme présenté ci-dessous.

Pour illustrer ce concept, nous utilisons lors des expérimentations des RBFs, dont la paramétrisation par primitives est décrite en §2.3.1.4. Un noyau d'interpolation de type plaque mince, décrit en §2.3.1.3, est utilisé. Pour les RBFs les primitives sont les centres de la transformation.

### 3.2.3 Enchaînement des déformations

#### 3.2.3.1 Principe de base

Afin de palier au problème de la composition, nous proposons l'enchaînement des déformations. Cette opération consiste à estimer la position des primitives de la fonction de déformation issue de l'enchaînement des deux autres.

Soient deux ensembles distincts de primitives contenus dans  $\mathbf{v}$  et  $\mathbf{v}'$ , représentant deux fonctions de déformation distinctes  $\mathcal{W}(\cdot; \mathbf{v})$  et  $\mathcal{W}(\cdot; \mathbf{v}')$ . L'objectif est d'estimer un troisième ensemble  $\mathbf{v}''$  pour que l'enchaînement des déformations induites par  $\mathbf{v}$  et  $\mathbf{v}'$  donne la fonction de déformation induite par  $\mathbf{v}''$  :  $\mathcal{W}(\cdot; \mathbf{v}'')$ .

Nous proposons une manière simple et efficace de réaliser cela, basée sur le concept de primitives. L'idée est d'appliquer la fonction de déformation induite par  $\mathbf{v}'$  à l'ensemble des primitives dans  $\mathbf{v}$ . Le résultat de l'opération constitue l'ensemble de primitives  $\mathbf{v}''$ . Ceci s'écrit :

$$\mathbf{v}'' = \mathcal{W}(\mathbf{v}; \mathbf{v}'), \quad (3.2)$$

où  $\mathcal{W}$  est appliquée à chaque primitive constituant  $\mathbf{v}$ . L'enchaînement de deux déformations  $\mathcal{W}(\cdot; \mathbf{v})$  et  $\mathcal{W}(\cdot; \mathbf{v}')$  est illustré sur la figure 3.2 et est par la suite noté  $\mathcal{W}(\cdot; \mathbf{v}') \square \mathcal{W}(\cdot; \mathbf{v})$ .

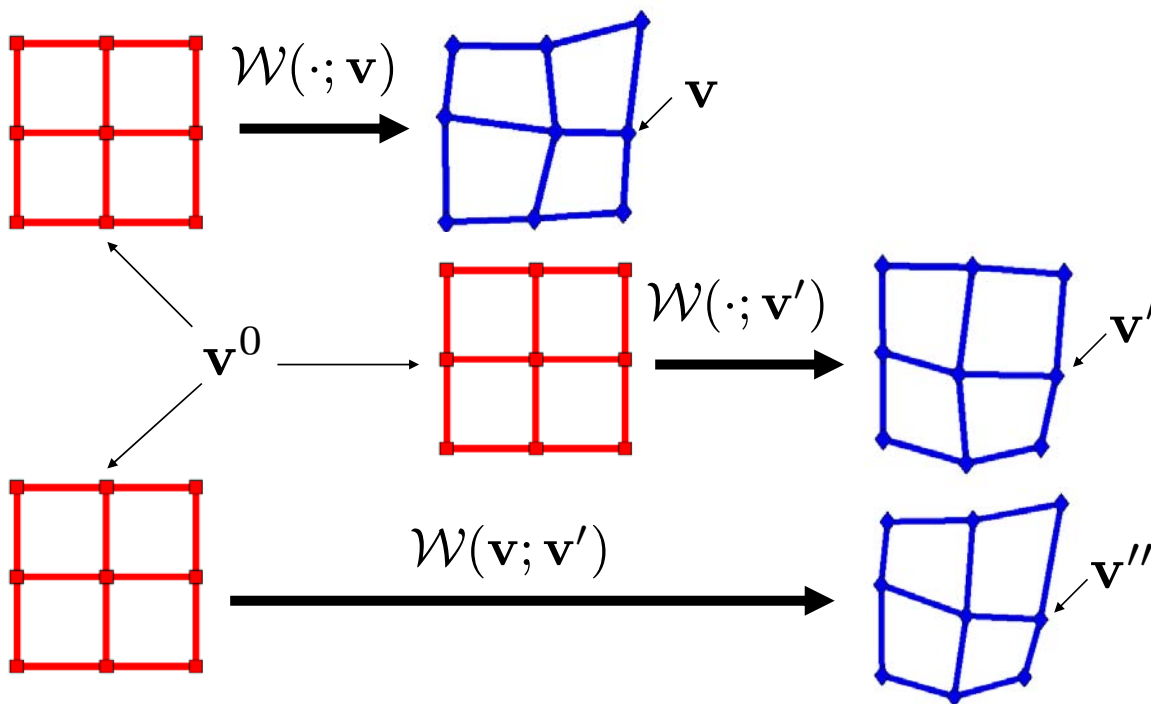


FIG. 3.2 – **Processus d'enchaînement des déformations.** Les primitives contenues dans  $\mathbf{v}''$  sont obtenues en enchaînant les fonctions de déformation engendrées par  $\mathbf{v}$  et  $\mathbf{v}'$  :  $\mathbf{v}'' = \mathcal{W}(\mathbf{v}; \mathbf{v}')$ .

Nous avons généré synthétiquement deux jeux de primitives  $\mathbf{v}$  et  $\mathbf{v}'$  en perturbant aléatoirement une grille régulière de taille  $3 \times 3$ . Les primitives  $\mathbf{v}''$  sont obtenues en enchaînant les fonctions de déformation engendrées par  $\mathbf{v}$  et  $\mathbf{v}'$ . La figure 3.3 illustre les résultats obtenus. On constate que la position des primitives  $\mathbf{v}''$  dépend de celles des primitives  $\mathbf{v}$  et  $\mathbf{v}'$ . La fonction de déformation induite  $\mathcal{W}(\cdot; \mathbf{v}'')$  est alors une combinaison de  $\mathcal{W}(\cdot; \mathbf{v}')$  et  $\mathcal{W}(\cdot; \mathbf{v})$ . Ainsi, l'enchaînement d'une déformation



$\mathcal{W}(\cdot; \mathbf{v})$  avec l'identité  $\mathcal{W}(\cdot; \mathbf{v}_0)$  retourne la déformation originelle  $\mathcal{W}(\cdot; \mathbf{v})$  :

$$\mathcal{W}(\cdot; \mathbf{v}) = \mathcal{W}(\cdot; \mathbf{v}) \square \mathcal{W}(\cdot; \mathbf{v}_0) \quad \text{et} \quad \mathcal{W}(\cdot; \mathbf{v}) = \mathcal{W}(\cdot; \mathbf{v}_0) \square \mathcal{W}(\cdot; \mathbf{v}).$$

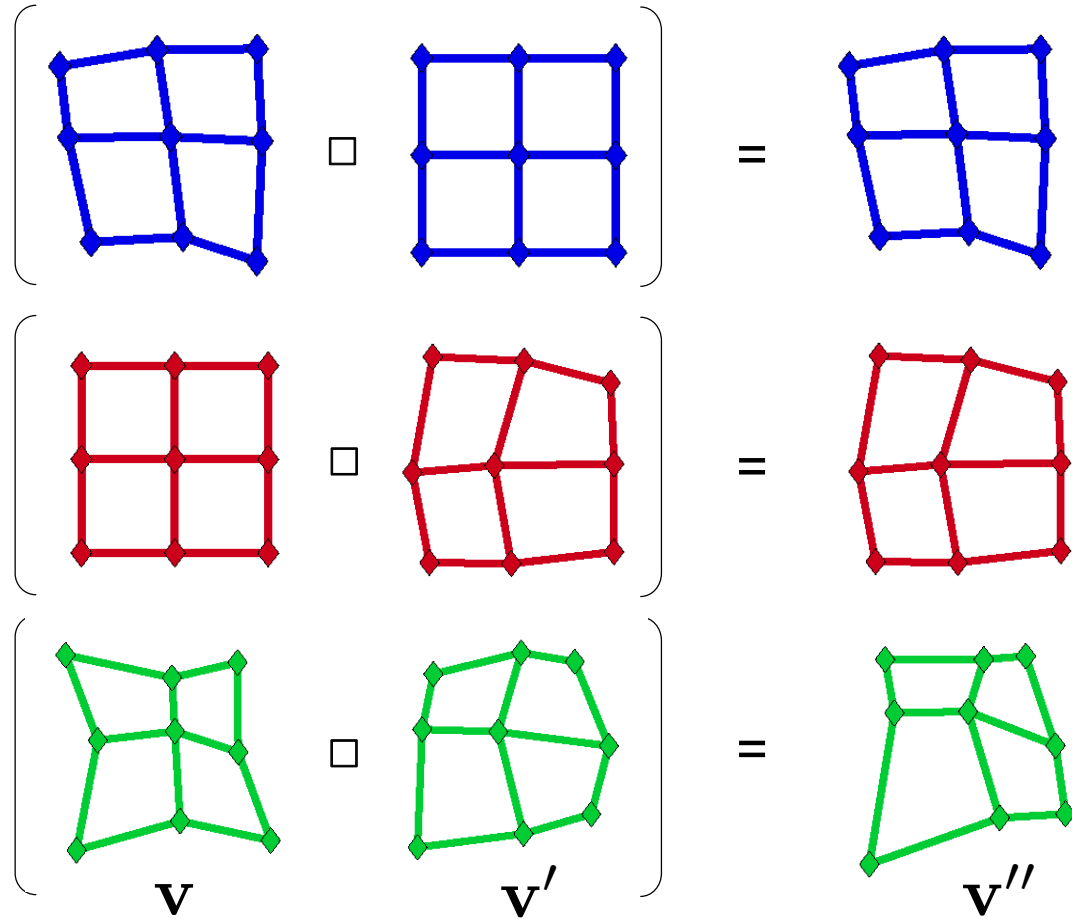


FIG. 3.3 – **Exemples d'enchaînement de déformations.** Les primitives contenues dans  $\mathbf{v}''$  sont issues du processus d'enchaînement des déformations engendrées par  $\mathbf{v}$  et  $\mathbf{v}'$ .

### 3.2.3.2 Estimation analytique pour les fonctions à base radiale

Pour des fonctions à base radiale, l'enchaînement de déformations conduit à un calcul analytique simple. Soit  $\mathcal{V}^0$  la matrice  $p \times 2$  contenant la position des  $p$  primitives sur l'image de référence (une ligne  $\mathbf{V}_i^0$  de  $\mathcal{V}^0$  correspond aux coordonnées  $x$  et  $y$  du  $i$ ème centre) et  $\mathcal{V}, \mathcal{V}', \mathcal{V}''$  les matrices contenant les  $p$  primitives engendrant respectivement les fonctions de déformation  $\mathcal{W}(\cdot; \text{vect}(\mathcal{V}))$ ,  $\mathcal{W}(\cdot; \text{vect}(\mathcal{V}'))$  et  $\mathcal{W}(\cdot; \text{vect}(\mathcal{V}''))$ . En combinant les équations (2.25) et (3.2) on obtient :

$$\mathbf{V}_i'' = \mathcal{W}(\mathbf{V}_i; \text{vect}(\mathcal{V}')) = \mathbf{o}_{\mathbf{V}_i}^\top \mathcal{E}_\lambda \mathcal{V}',$$

avec  $\mathcal{E}_\lambda$  défini par l'équation (2.24) et ne dépend que des primitives dans  $\mathcal{V}^0$ , et

$$\mathbf{o}_{\mathbf{V}_i}^\top = \left( \phi(d^2(\mathbf{V}_i, \mathbf{V}_1^0)) \quad \cdots \quad \phi(d^2(\mathbf{V}_i, \mathbf{V}_p^0)) \quad \mathbf{V}_i^\top \quad 1 \right).$$

En appliquant l'équation ci-dessus à chaque primitive contenue dans  $\mathcal{V}$ , on obtient :

$$\mathcal{V}'' = O_{\mathcal{V}} \mathcal{E}_{\lambda} \mathcal{V}',$$

avec

$$O_{\mathcal{V}} = \begin{pmatrix} \mathbf{o}_{\mathbf{v}_1}^{\top} \\ \mathbf{o}_{\mathbf{v}_2}^{\top} \\ \vdots \\ \mathbf{o}_{\mathbf{v}_p}^{\top} \end{pmatrix}.$$

Le processus d'enchaînement revient, pour des fonctions à base radiale, à multiplier 3 matrices entre elles dont les tailles respectives sont  $p \times p + 3$ ,  $p + 3 \times p$  et  $p \times 2$ . Notons que la matrice  $\mathcal{E}_{\lambda}$  est constante.

### 3.2.4 Renversement d'une déformation

#### 3.2.4.1 Principe de base

Le processus de renversement d'une déformation approxime l'inversion de fonction. L'idée est d'estimer directement la position des primitives de la fonction de déformation « inversée ».

Soit un ensemble de primitives contenu dans  $\mathbf{v}$ . L'objectif est d'estimer l'ensemble de primitives  $\mathbf{v}'$  défini tel que la fonction de déformation induite par  $\mathbf{v}'$  produise un déplacement « inverse » de celui engendré par  $\mathbf{v}$ . Comme pour l'enchaînement, l'approche par primitives permet de résoudre ce problème très simplement. L'idée est que si l'on applique la fonction de déformation induite par les primitives  $\mathbf{v}'$  à l'ensemble des primitives  $\mathbf{v}$ , les primitives de référence  $\mathbf{v}^0$  doivent être retrouvées, comme l'illustre la figure 3.4. En d'autres termes, l'enchaînement d'une fonction de déformation avec sa fonction de déformation renversée engendre la transformation identité. Ceci s'écrit :

$$\mathcal{W}(\mathbf{v}; \mathbf{v}') = \mathbf{v}^0, \quad (3.3)$$

où  $\mathcal{W}$  est appliquée à chaque primitive contenue dans  $\mathbf{v}$ . La solution est obtenue simplement pour les fonctions à base radiale, en résolvant un système linéaire<sup>5</sup> dont le nombre d'équations et d'inconnues correspondent au nombre de primitives. Notons que suivant la fonction de déformation utilisée, il peut être plus judicieux de résoudre le problème suivant pour pouvoir évaluer les primitives  $\mathbf{v}'$  :  $\mathcal{W}(\mathbf{v}'; \mathbf{v}) = \mathbf{v}^0$ . Le renversement d'une déformation  $\mathcal{W}(\cdot; \mathbf{v})$  est par la suite notée  $\mathcal{W}(\cdot; \mathbf{v})^{\circ}$ .

Afin d'illustrer le processus de renversement, nous avons généré synthétiquement un jeu de primitives  $\mathbf{v}$  en perturbant aléatoirement une grille régulière  $\mathbf{v}^0$  de  $3 \times 3$  primitives. Les primitives  $\mathbf{v}'$  sont obtenues par le processus de renversement appliqué à la fonction de déformation  $\mathcal{W}(\cdot; \mathbf{v})$ . Ensuite les fonctions de déformation  $\mathcal{W}(\cdot; \mathbf{v})$  et  $\mathcal{W}(\cdot; \mathbf{v}')$  sont enchaînées, un nouveau jeu de primitives  $\mathbf{v}''$  est alors engendré. Ces dernières sont équivalentes aux primitives de référence  $\mathbf{v}^0$ , l'erreur résiduelle moyenne est très faible. La figure 3.5 illustre les résultats obtenus.

#### 3.2.4.2 Estimation analytique pour les fonctions à base radiale

Nous reprenons ici les mêmes notations que pour l'enchaînement des déformations décrit en §3.2.3.2. On cherche les primitives issues du renversement de  $\mathcal{W}(\cdot; \text{vect}(\mathcal{V}))$ . En combinant les équations (2.25) et (3.3), on obtient :

$$\mathcal{V}^0 = O_{\mathcal{V}} \mathcal{E}_{\lambda} \mathcal{V}', \quad (3.4)$$

<sup>5</sup>La procédure décrite en §1.6.1 est utilisée.

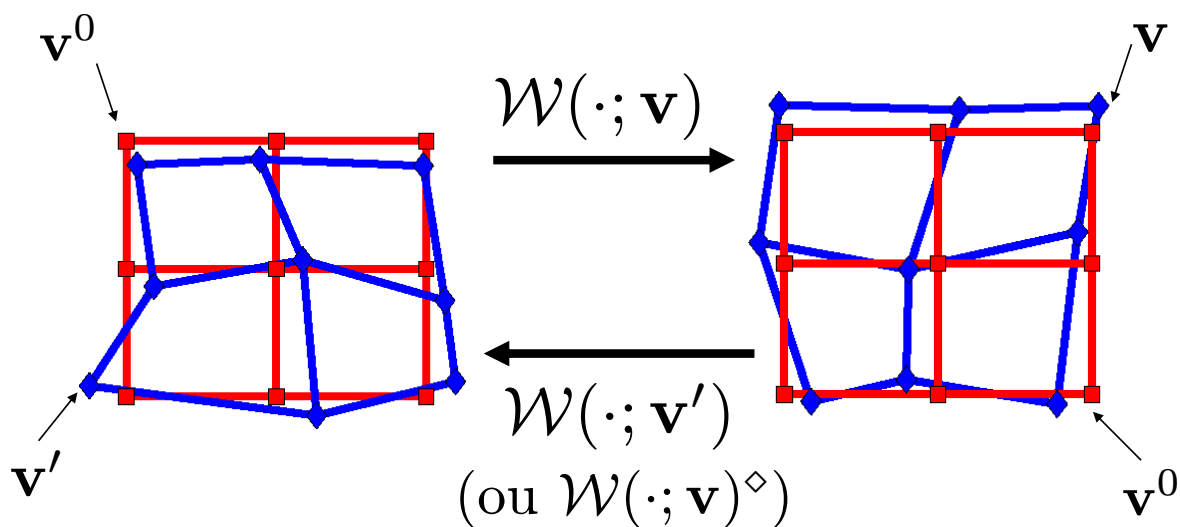


FIG. 3.4 – **Processus de renversement d’une déformation.** Les primitives contenues dans  $\mathbf{v}'$  sont définies par les équations  $\mathcal{W}(\mathbf{v}; \mathbf{v}') = \mathbf{v}^0$  ou  $\mathcal{W}(\mathbf{v}'; \mathbf{v}) = \mathbf{v}^0$ .

La matrice  $\mathcal{V}'$  contenant les primitives engendrant la fonction de déformation renversée est donnée par :

$$\mathcal{V}' = (\mathcal{O}_{\mathcal{V}} \mathcal{E}_{\lambda})^{-1} \mathcal{V}^0, \quad (3.5)$$

où la matrice  $\mathcal{O}_{\mathcal{V}} \mathcal{E}_{\lambda}$  est de taille  $p \times p$ .

### 3.2.5 Recalage compositionnel guidé par primitives

Grâce aux propriétés de la paramétrisation par primitives, nous étendons les algorithmes compositionnels, notamment l’algorithme IC-GN, à des fonctions de déformation ne formant pas un groupe. Les trois étapes suivantes sont répétées jusqu’à convergence :

- ▷ **Etape 1 : Recalage global.** Les primitives courantes  $\mathbf{u}$  sont utilisées pour recalculer  $\mathcal{I}$  sur l’image de texture :

$$\mathcal{I}_W(\mathbf{q}) = \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u})).$$

- ▷ **Etape 2 : Recalage local.** Les déplacements  $\Delta \mathbf{u}$  des primitives de référence sont estimés sur l’image  $\mathcal{I}_W$ . Plusieurs approximations de la fonction de coût décrites en §1.6.2 peuvent être utilisées. Notons que pour les algorithmes compositionnels inverses, comme par exemple IC-GN, le renversement de la fonction de déformation locale  $\mathcal{W}(\cdot; \mathbf{u}_0 + \Delta \mathbf{u})$  est réalisé à cette étape en utilisant l’équation (3.3).
- ▷ **Etape 3 : Mise à jour.** Les fonctions de déformation induites respectivement par  $\mathbf{u}$  et  $\mathbf{u}_0 + \Delta \mathbf{u}$  sont enchaînées en utilisant l’équation (3.2), permettant la mise à jour des primitives  $\mathbf{u}$  sur l’image courante.

Une vision globale du recalage compositionnel ainsi qu’une illustration de l’algorithme IC-GN guidé par primitives sont présentées en figure 3.6. Ses différentes étapes sont récapitulées sur le tableau 3.1. Les différences avec l’algorithme IC-GN « classique », introduit en §2.2.2.3, interviennent au niveau de l’étape de mise à jour.

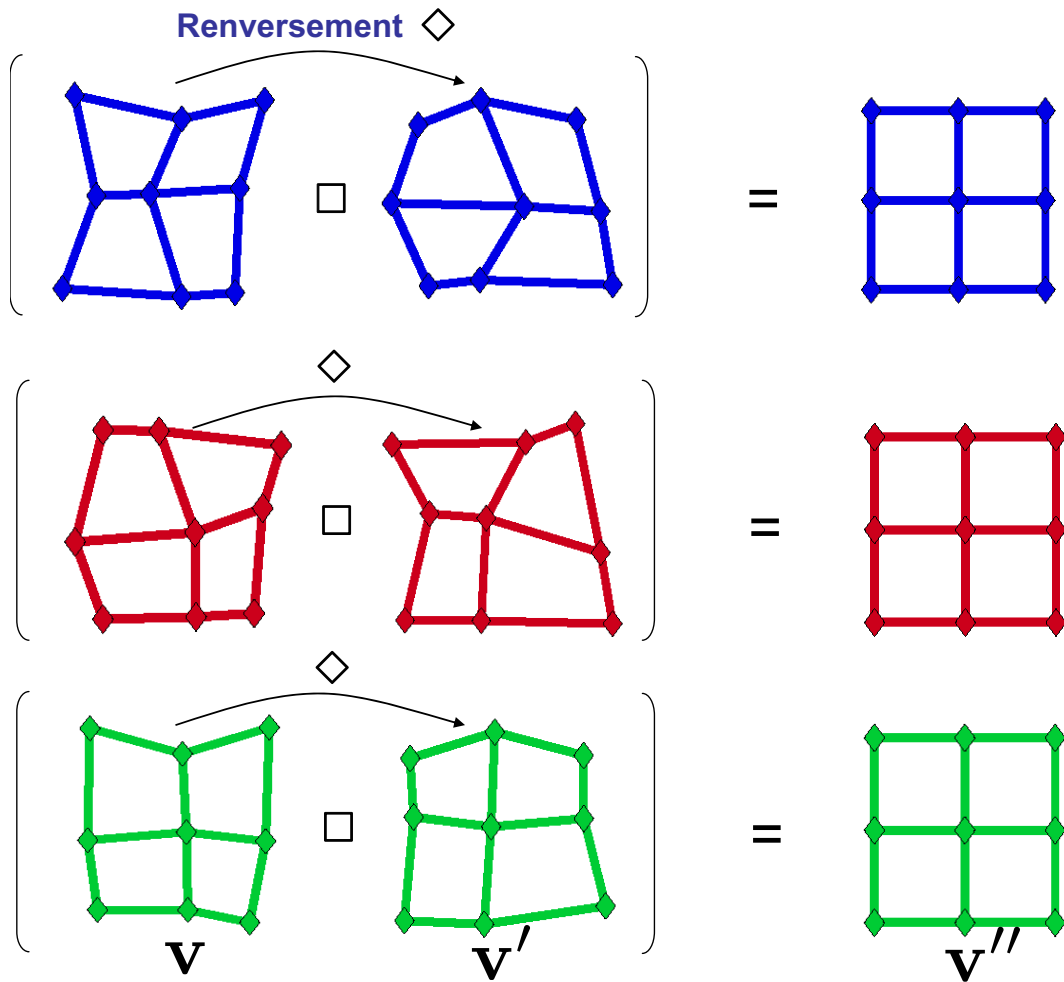


FIG. 3.5 – **Exemples de renversement d’une déformation.** Les primitives contenues dans  $\mathbf{v}'$  sont issues du renversement de  $\mathcal{W}(\cdot; \mathbf{v})$ . Notons qu’à l’issue de l’enchaînement des déformations  $\mathcal{W}(\cdot; \mathbf{v})$  et  $\mathcal{W}(\cdot; \mathbf{v}')$  les primitives de référence sont bien retrouvées :  $\|\mathbf{v}'' - \mathbf{v}^0\| \approx 0$ .

<b>Hors ligne</b>
<ul style="list-style-type: none"> <li>▷ Évaluer le gradient de l'image de référence <math>\nabla \mathcal{I}_0</math> à <math>\mathcal{W}(\mathbf{q}; \mathbf{u}_0) = \mathbf{q}</math>.</li> <li>▷ Évaluer le gradient de la fonction de déformation <math>\frac{\partial \mathcal{W}}{\partial \mathbf{u}}</math> à <math>(\mathbf{q}; \mathbf{u}_0)</math>.</li> <li>▷ Calculer les vecteurs gradients <math>\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0) = \nabla \mathcal{I}_0^\top  _{\mathcal{W}(\mathbf{q}; \mathbf{u}_0)} \frac{\partial \mathcal{W}}{\partial \mathbf{u}}  _{(\mathbf{q}; \mathbf{u}_0)}</math>.</li> <li>▷ Former la matrice Jacobienne <math>\mathcal{J}_0</math> en rassemblant les vecteurs gradients <math>\mathbf{g}_0^\top(\mathbf{q}; \mathbf{u}_0)</math>.</li> <li>▷ Calculer la pseudo inverse de <math>\mathcal{J}_0</math> : <math>\mathcal{J}_0^\dagger</math>.</li> </ul>
<b>En ligne</b>
<ul style="list-style-type: none"> <li>▷ Recaler par <math>\mathcal{W}(\mathbf{q}; \mathbf{u})</math> l'image courante sur l'image de référence : <math>\mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))</math>.</li> <li>▷ Calculer l'image de différence <math>\mathcal{D}(\mathbf{q}) = \mathcal{I}_0(\mathbf{q}) - \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))</math>.</li> <li>▷ Calculer l'incrément <math>\Delta \mathbf{u} = -\mathcal{J}_0^\dagger \mathcal{D}</math>.</li> <li>▷ Renversement de la fonction de déformation <math>\mathcal{W}(\cdot; \mathbf{u}_0 + \Delta \mathbf{u})</math>, les primitives <math>\mathbf{u}_0 + \Delta \mathbf{u}_{inv}</math> engendrant la fonction de déformation renversée sont données par : <math>\mathcal{W}(\mathbf{u}_0 + \Delta \mathbf{u}; \mathbf{u}_0 + \Delta \mathbf{u}_{inv}) = \mathbf{u}_0</math>.</li> <li>▷ Enchaînement des déformations <math>\mathcal{W}(\cdot; \mathbf{u})</math> et <math>\mathcal{W}(\cdot; \mathbf{u}_0 + \Delta \mathbf{u}_{inv})</math>, les primitives <math>\mathbf{u}</math> sont mises à jour par : <math>\mathbf{u} \leftarrow \mathcal{W}(\mathbf{u}_0 + \Delta \mathbf{u}_{inv}; \mathbf{u})</math>.</li> </ul>

TAB. 3.1 – Les différentes étapes d'une itération de l'algorithme IC-GN guidé par primitives.

Notons que pour les fonctions à base radiale, l'étape supplémentaire introduite dans (Matthews and Baker, 2004; Romdhani and Vetter, 2003), et donnée par l'équation (3.1), n'est pas requise. Les processus d'enchaînement et de renversement de déformations prennent directement en compte les contraintes relatives à la fonction de déformation.

## 3.3 Recalage basé apprentissage

### 3.3.1 Introduction

Le principe des méthodes basées apprentissage est d'apprendre la relation  $\Delta \mathbf{u} = g(\mathcal{D})$  entre la variation d'intensité  $\mathcal{D}$  (c'est-à-dire l'image de différence) et la variation locale des paramètres  $\Delta \mathbf{u}$ . Elle est apprise sur des données générées synthétiquement par perturbation aléatoire des paramètres de référence  $\mathbf{u}_0$ . Différentes méthodes d'apprentissage peuvent être utilisées. La plus commune est la régression de type moindres carrés (Cootes et al., 1998; Jurie and Dhome, 2002a,b; Gay-Bellile et al., 2007a, 2006). La relation engendrée est linéaire : elle est représentée par une *matrice d'interaction*  $\mathcal{G}$  :

$$\Delta \mathbf{u} = \mathcal{G} \mathcal{D}.$$

D'autres méthodes utilisent des fonctions noyau pour l'apprentissage : régression par SVM<sup>6</sup> ou RVM<sup>7</sup> (Agarwal and Triggs, 2006; Chateau et al., 2008). La relation obtenue est alors non-linéaire. Ces méthodes d'apprentissage ne sont pas étudiées dans ce chapitre, néanmoins leur comparaison avec l'approche proposée est à envisager. Nous nous concentrons sur l'apprentissage de type moindres carrés. L'approximation linéaire « classique » présente certaines limitations décrites en §3.3.2.2. Un compromis doit notamment être trouvé entre précision et largeur du bassin de convergence. Afin de palier

<sup>6</sup>"Support Vector Machine".

<sup>7</sup>"Relevance Vector Machine".

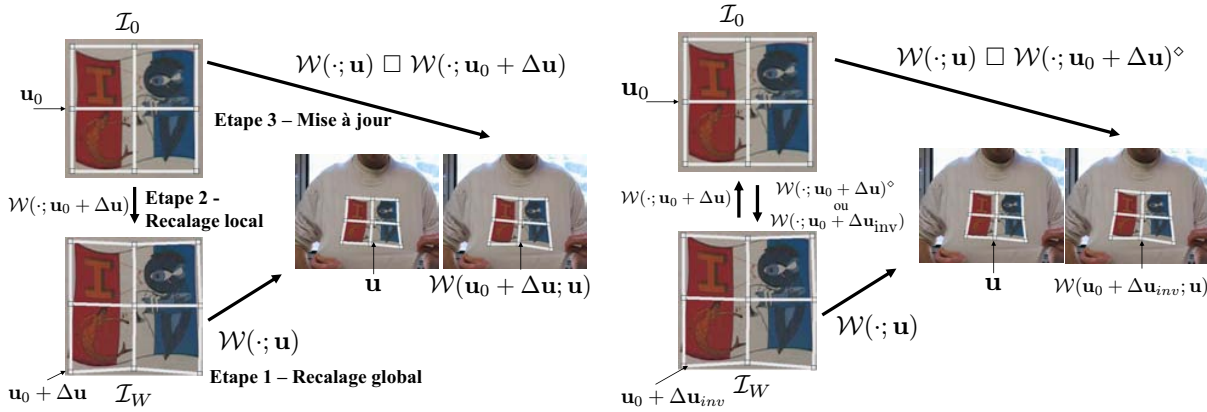


FIG. 3.6 – **Recalage compositionnel guidé par primitives.** A gauche : les étapes du recalage compositionnel guidé par primitives. A droite : principe de l’algorithme IC-GN guidé par primitives.

à cet inconvénient, nous proposons une relation linéaire par morceaux. Plusieurs matrices d’interaction « spécialisées » sur un domaine restreint d’amplitude de déplacement sont utilisées. Différentes possibilités sont proposées et comparées pour combiner ces différentes matrices d’interaction.

Une fois la relation  $g$  apprise, elle est utilisée lors du recalage pour mettre à jour les paramètres courants. Elle n’est cependant valide que localement, c’est-à-dire autour des paramètres de référence  $\mathbf{u}_0$ , nécessitant par conséquent l’emploi d’algorithmes compositionnels. Le recalage d’images basé apprentissage se limite théoriquement à des transformations formant un groupe, comme par exemple une homographie (Chateau et al., 2008; Jurie and Dhome, 2002a,b). Dans (Cootes et al., 1998), un algorithme basé apprentissage combiné avec un modèle actif est néanmoins proposé. Les auteurs supposent que la relation apprise  $g$  peut être appliquée directement aux paramètres courants et utilise par conséquent une mise à jour additive des paramètres. L’étape complexe de composition est ainsi évitée. Cette hypothèse n’est cependant pas valable en pratique et les performances de convergence s’en trouvent être fortement dégradées.

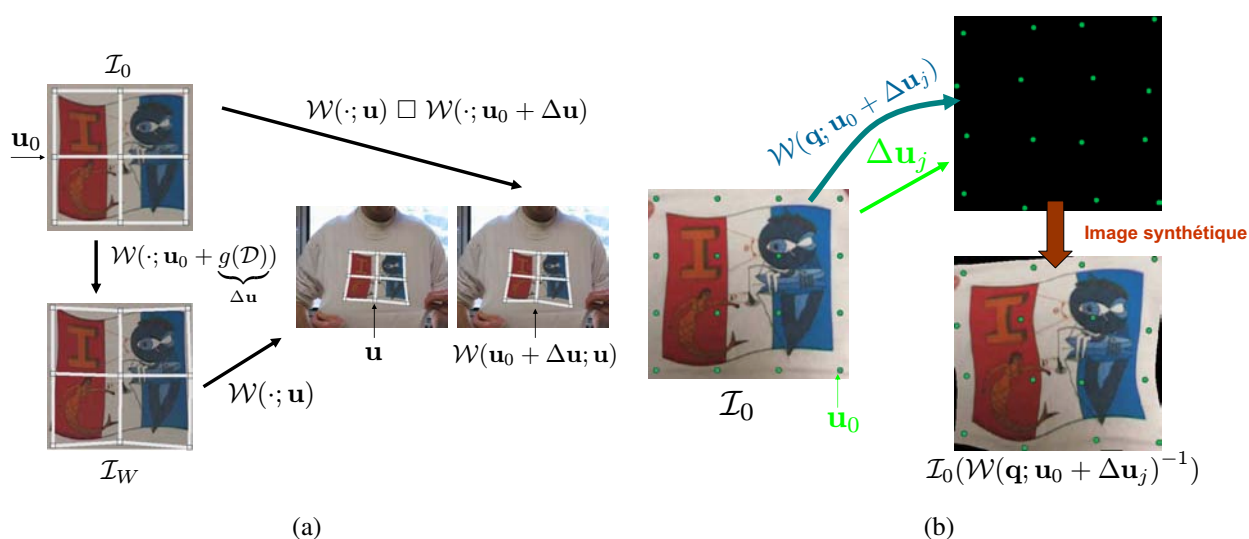
Nous avons vu en §3.2 que l’algorithme IC-GN guidé par primitives est compatible avec certaines fonctions de déformation. Le processus d’enchaînement de déformations peut également être utilisé pour étendre les algorithmes basés apprentissage aux modèles déformables. La mise à jour se fait par enchaînement de la fonction de déformation courante  $\mathcal{W}(\cdot; \mathbf{u})$  avec celle obtenue lors du recalage local par apprentissage :  $\mathcal{W}(\cdot; \mathbf{u}_0 + g(\mathcal{D}))$ , avec  $\mathcal{D} = \mathcal{I}_0 - \mathcal{I}_W$ . Notons que le processus de renversement n’est pas utilisé au cours des itérations puisque  $g$  recalc l’image  $\mathcal{I}_W$  sur l’image de texture  $\mathcal{I}_0$  (sens direct). Il est tout de même requis lors de l’apprentissage comme présenté en §3.3.2.1. Dans la suite du tapuscrit, l’algorithme compositionnel basé apprentissage est appelé DC-Le "Direct Compositionnal Learning". Pour des fonctions de déformation, on parle d’algorithme DC-Le guidé par primitives. La figure 3.7(a) illustre son principe. Le tableau 3.2 récapitule les étapes de cet algorithme.

### 3.3.2 Approximation linéaire

Cette approximation de la relation entre la variation d’intensité et la variation locale des paramètres a été utilisée par de nombreux auteurs, notamment dans (Cootes et al., 1998; Jurie and Dhome, 2002a). Après avoir brièvement introduit la relation linéaire modélisée par une matrice d’interaction, nous mettons en évidence ses limitations.

<b>Hors ligne</b>
▷ Apprendre la relation $g$ (§3.3.3).
<b>En ligne</b>
▷ Recaler par $\mathcal{W}(\mathbf{q}; \mathbf{u})$ l'image courante sur l'image de référence : $\mathcal{I}_{\mathcal{W}}(\mathbf{q}) = \mathcal{I}(\mathcal{W}(\mathbf{q}; \mathbf{u}))$ .
▷ Calculer l'image de différence $\mathcal{D} = \mathcal{I}_0 - \mathcal{I}_{\mathcal{W}}$ .
▷ Calculer l'incrément local $\Delta \mathbf{u} = g(\mathcal{D})$ .
▷ Enchaînement des déformations $\mathcal{W}(\cdot; \mathbf{u})$ et $\mathcal{W}(\cdot; \mathbf{u}_0 + \Delta \mathbf{u})$ , les primitives dans $\mathbf{u}$ sont mises à jour par : $\mathbf{u} \leftarrow \mathcal{W}(\mathbf{u}_0 + \Delta \mathbf{u}; \mathbf{u})$ .

TAB. 3.2 – Les différentes étapes d'une itération de l'algorithme DC-Le guidé par primitives.

FIG. 3.7 – **Principe du recalage d'images basé apprentissage.** (a) Principe de l'algorithme DC-Le guidé par primitives. (b) Création d'un exemple d'apprentissage pour des fonctions de déformation guidées par primitives.

### 3.3.2.1 Apprendre une matrice d'interaction

**Génération des exemples d'apprentissage.** Une matrice d'interaction  $\mathcal{G}$  est apprise à partir d'images synthétiques  $\{\mathcal{I}_j\}_{j=1}^p$  issues de la déformation de l'image de texture. Les images synthétiques sont obtenues en perturbant aléatoirement les paramètres de référence  $\mathbf{u}_0$ . Ces perturbations engendrent des déplacements compris entre une borne inférieure  $\gamma_i$  et une borne supérieure  $\gamma_s$  exprimées en pixels. Elles déterminent le domaine de validité de la matrice d'interaction. Notons que fixer ces bornes peut parfois s'avérer complexe lorsque les paramètres sont difficilement interprétables comme c'est le cas pour les fonctions à base radiale. Plusieurs vecteurs de paramètres  $\{\mathbf{u}_j\}_{j=1}^p$  sont obtenus, avec  $\mathbf{u}_j \leftarrow \mathbf{u}_0 + \Delta \mathbf{u}_j$ . Les images synthétiques associées peuvent être générées à partir de la fonction de déformation  $\mathcal{W}(\mathbf{q}; \mathbf{u}_j)$  transférant un pixel de l'image de référence sur l'image  $\mathcal{I}_j$ . Chaque pixel doit alors être traité séparément, par minimisation d'un critère non-linéaire :

$$\mathcal{I}_j(\mathbf{q}) \leftarrow \mathcal{I}_0(\arg \min_{\mathbf{q}} \|\mathcal{W}(\mathbf{q}; \mathbf{u}_j) - \mathbf{q}\|).$$

Il en résulte un procédé extrêmement coûteux en terme de temps de calcul. Nous privilégions l'emploi de la fonction de déformation inverse  $\mathcal{W}(\mathbf{q}; \mathbf{u}_j)^{-1}$ , transférant un pixel de l'image  $\mathcal{I}_j$  vers l'image de référence :

$$\mathcal{I}_j(\mathbf{q}) \leftarrow \mathcal{I}_0(\mathcal{W}(\mathbf{q}; \mathbf{u}_j)^{-1}). \quad (3.6)$$

L'image synthétique est alors texturée par simple interpolation de l'image de référence.

**Application à la paramétrisation par primitives.** Les positions des primitives sur l'image de texture sont perturbées par sélection aléatoire de directions  $\{\boldsymbol{\theta}_j\}_{j=1}^p$  et d'amplitudes de déplacement  $\{\mathbf{r}_j\}_{j=1}^p : \Delta \mathbf{u}_j = \begin{pmatrix} \mathbf{r}_j \odot \cos(\boldsymbol{\theta}_j) \\ \mathbf{r}_j \odot \sin(\boldsymbol{\theta}_j) \end{pmatrix}$ . Les bornes de perturbation des paramètres  $\{\Delta \mathbf{u}_j\}_{j=1}^p$  sont fixées naturellement puisque les primitives sont exprimées en pixel. Les images synthétiques sont générées en utilisant l'équation (3.6). Le processus de renversement de déformation donné en §3.2.4 est alors utilisé pour approximer l'inversion de  $\mathcal{W}(\mathbf{q}; \mathbf{u}_j)$ . La création d'une image synthétique  $\mathcal{I}_j$  pour des fonctions de déformation guidées par primitives est illustrée sur la figure 3.7(b).

**Apprentissage par régression de type « moindres carrés ».** Les vecteurs d'erreurs résiduelles sont donnés par :

$$\mathbf{D}_j = \text{vect}(\mathcal{I}_0 - \mathcal{I}_j),$$

et les exemples d'apprentissage sont rassemblés dans les matrices  $\delta \mathcal{U} = \begin{pmatrix} \Delta \mathbf{u}_1 & \cdots & \Delta \mathbf{u}_p \end{pmatrix}$  et  $\delta \mathcal{D} = \begin{pmatrix} \mathbf{D}_1 & \cdots & \mathbf{D}_p \end{pmatrix}$ . Nous cherchons la matrice d'interaction  $\mathcal{G}$  tel que  $\delta \mathcal{U} = \mathcal{G} \delta \mathcal{D}$ . Une solution possible est obtenue en minimisant un critère de type moindres carrés dans l'espace des paramètres (exprimés en pixels dans le cadre de la paramétrisation par primitives) donné par :  $\|\mathcal{G} \delta \mathcal{D} - \delta \mathcal{U}\|^2$ . Nous privilégions par la suite l'apprentissage de matrices d'interaction dans l'espace d'intensité (ou de couleur) exprimé en unité d'intensité.

$$\min_{\mathcal{G}} \|\delta \mathcal{D} - \mathcal{G}^\dagger \delta \mathcal{U}\|^2 \Rightarrow \mathcal{G} = \left( \delta \mathcal{D} \delta \mathcal{U}^\top (\delta \mathcal{U} \delta \mathcal{U}^\top)^{-1} \right)^\dagger.$$

Nous avons observé expérimentalement en §3.4 que la deuxième méthode donne de meilleurs résultats, étant notamment beaucoup plus résistante au bruit.

### 3.3.2.2 Limitations de l'approximation linéaire

Le domaine de validité de l'approximation linéaire est conditionné par la borne supérieure  $\gamma_s$  des perturbations lors de l'apprentissage<sup>8</sup>. Pour des grandes valeurs de  $\gamma_s$ , la relation linéaire est valide sur un large domaine de déplacement, il en résulte un bassin de convergence élargi comme l'illustre la figure 3.8(a). En contrepartie, les petits déplacements sont mal « expliqués », entraînant une perte de précision par rapport à une relation apprise avec une petite valeur de  $\gamma_s$ , voir figure 3.8(b). Les performances de la relation linéaire dépendent fortement de  $\gamma_s$  : celui-ci contrôle le compromis entre précision et largeur du bassin de convergence.

Les résultats expérimentaux présentés sur les figures 3.8(a) et 3.8(b) illustrent ce phénomène. Ils sont obtenus en suivant la procédure décrite en §3.4.1.

<sup>8</sup>La borne inférieure est fixée à  $\gamma_i = 0$ .



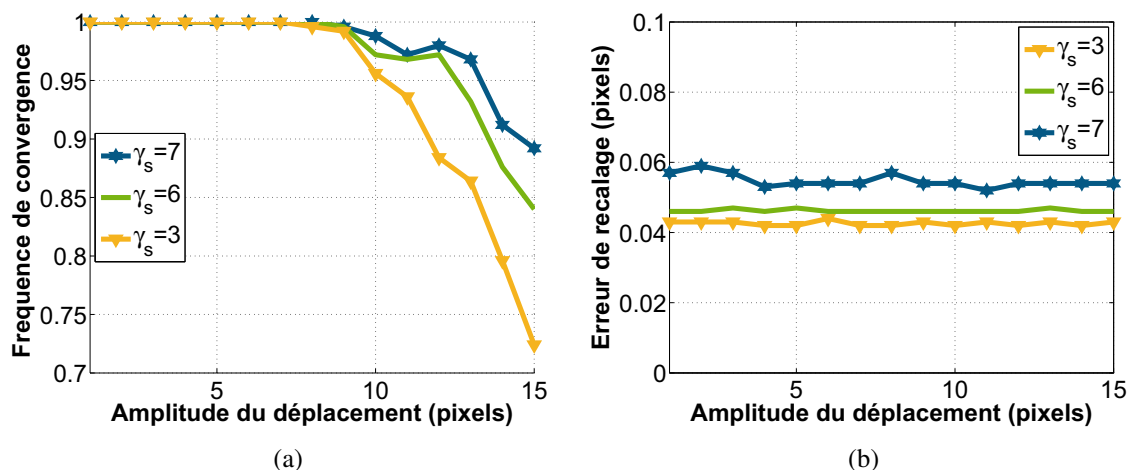


FIG. 3.8 – **Limitation de l'approximation linéaire.** Les performances de l'approche par apprentissage linéaire sont testées en fonction de l'amplitude  $\gamma_s$  des déformations ayant engendrées les exemples d'apprentissage. (a) Performance en terme de fréquence de convergence lorsque l'amplitude de la déformation varie. (b) Performance en terme de précision lorsque l'amplitude de la déformation varie. Le processus expérimental est donné en §3.4.1.

### 3.3.3 Approximation linéaire par morceaux

#### 3.3.3.1 Présentation

Afin de palier aux limitations de l'approximation linéaire révélées ci-dessus, nous proposons d'apprendre plusieurs matrices d'interaction  $\mathcal{G}_1, \dots, \mathcal{G}_l$ , sur des domaines de perturbation dont l'amplitude décroît, la relation  $\mathcal{G}_1$  possédant le plus grand domaine de validité. Elles forment ainsi une approximation linéaire par morceaux de la fonction de coût. Différentes possibilités existent pour combiner les matrices  $\{\mathcal{G}_i\}_{i=1}^l$ . Il en résulte différentes relations linéaires par morceaux.

#### 3.3.3.2 Différentes relations linéaires par morceaux

**Les approches bouclées.** Ce sont des approches plutôt naïves. Une première possibilité est d'appliquer toutes les matrices en boucle (LOOP). La relation  $\mathcal{G}_1$  est utilisée en premier (elle correspond aux forts déplacements) jusqu'à convergence. Ensuite, les  $\{\mathcal{G}_i\}_{i=2}^{l-1}$  sont appliquées les unes à la suite des autres. A la fin du processus, la relation  $\mathcal{G}_l$  apprise sur de tous petits déplacements garantit la précision du recalage. L'autre possibilité revient à tester toutes les relations à chaque itération. Celle qui introduit la réduction de l'erreur la plus importante est conservée (BEST). Le principal inconvénient de ces méthodes est qu'elles ne sont pas efficaces. En effet, LOOP introduit un grand nombre d'itérations tandis que BEST implique un coût élevé pour une itération. Ceci est d'autant plus vrai que le nombre de matrices d'interaction  $l$  est grand.

**La sélection statistique.** L'idée est de sélectionner la matrice la plus adaptée à chaque itération (PROB). Chacune d'entre elles possède un domaine spécifique de validité en terme d'amplitude de déplacement. Cette amplitude ne peut pas être déterminée *a priori* lors du recalage. Nous proposons d'apprendre la relation entre l'erreur d'intensité et les intervalles d'amplitude de déplacement. A chaque matrice d'interaction est associée une densité de probabilité  $P(\mathcal{G}_i | e(\mathcal{D}))$ , avec

$e(\mathcal{D}) = \text{rms}(\text{vect}(\mathcal{D}))$  définissant l'erreur d'intensité pour un vecteur résiduel  $\text{vect}(\mathcal{D})$ . Ces densités sont apprises sur les exemples d'apprentissage ayant engendrés les  $\{\mathcal{G}_i\}_{i=1}^l$ . En pratique, une approximation Gaussienne s'avère être proche des distributions réelles. La sélection de la matrice d'interaction la plus appropriée étant donné l'erreur en intensité courante se fait simplement par :

$$j = \arg \max_i P(\mathcal{G}_i | e(\mathcal{D})). \quad (3.7)$$

**La combinaison ou mixture de matrices.** Ces méthodes combinent les matrices d'interaction par un modèle de mixture probabiliste :

$$\mathcal{G} \leftarrow \sum_{i=1}^l a_i \mathcal{G}_i \quad \text{avec} \quad \sum_{i=1}^l a_i = 1 \quad \text{et} \quad \{a_i \geq 0\}_{i=1}^l.$$

Nous comparons par la suite deux choix pour les proportions des mixtures  $\{a_i\}_{i=1}^l$ . Une première approche (CMM) affecte la même probabilité  $a_i = \frac{1}{p}$  à chaque matrice d'interaction tandis que la deuxième (GMM) les pondère par les densités de probabilité  $P(\mathcal{G}_i | e(\mathcal{D}))$  normalisées :

$$a_i = \frac{P(\mathcal{G}_i | e(\mathcal{D}))}{\sum_{k=1}^l P(\mathcal{G}_k | e(\mathcal{D}))}.$$

Notons que les relations linéaires par morceaux décrites ci-dessus se terminent en appliquant, jusqu'à convergence, la matrice d'interaction  $\mathcal{G}_l$ . Cette étape supplémentaire garantit un recalage précis.

## 3.4 Résultats expérimentaux

### 3.4.1 Comparaison des différentes méthodes d'apprentissage

Afin de comparer les performances des différentes relations linéaires par morceaux décrites en §3.3.3.2, une transformation de type homographie est utilisée. Ainsi, aucune approximation n'est faite lors de l'apprentissage (inversion) et également lors du recalage (composition). Une paramétrisation de l'homographie par 4 points (ou primitives) est utilisée. Nous utilisons 6 matrices d'interactions pour définir les relations linéaires par morceaux. Les bornes de déplacement  $\gamma_i$  et  $\gamma_s$ , associées à chaque matrice d'interaction sont données dans le tableau 3.3.

	matrice $\mathcal{G}_1$	matrice $\mathcal{G}_2$	matrice $\mathcal{G}_3$	matrice $\mathcal{G}_4$	matrice $\mathcal{G}_5$	matrice $\mathcal{G}_6$
Borne inférieure $\gamma_i$	0	0	0	0	0	0
Borne supérieure $\gamma_s$	20	15	10	7	4	2

TAB. 3.3 – Les bornes de déplacement utilisées pour apprendre les différentes matrices d'interaction. Les mêmes valeurs sont utilisées pour les expérimentations en §3.4.2.

Les différentes relations linéaires par morceaux sont comparées dans des conditions contrôlées : des images synthétiques sont générées à partir de l'image de texture.

**Protocole.** Les positions des primitives sont aléatoirement perturbées avec une amplitude  $r$ . Un bruit Gaussien est également ajouté sur l'image synthétisée. La variance  $\sigma$  associée est un pourcentage sur l'intensité maximale en niveau de gris. Ces deux paramètres sont modifiés indépendamment avec comme valeur par défaut  $r = 6$  pixels et  $\sigma = 1\%$ . La qualité du recalage est évaluée en calculant la

distance Euclidienne moyenne entre les primitives ayant induit l'image synthétique et celles estimées. La convergence vers la solution recherchée est déclarée lorsque le score obtenu est inférieur à 1 pixel. Les résultats présentés sont des moyennes sur 500 essais.

**Fréquence de convergence.** (figures 3.9(a) et 3.9(b)). C'est le pourcentage de convergence vers la solution recherchée. Par rapport aux déplacements, les différentes relations ont des performances comparables. En effet, leur fréquence de convergence est supérieure à 95% jusqu'à 20 pixels de déplacement. A partir de 25 pixels, elles sont toutes aux alentours de 65% de recalage réussi. La relation CMM apparaît avoir un bassin de convergence très légèrement plus faible. Notons que la relation linéaire par morceaux PROB présente quelques cas d'échecs pour des amplitudes de déplacement comprises entre 6 et 8 pixels. Les relations linéaires par morceaux basées sur les densités de probabilité  $P(\cdot|e(\mathcal{D}))$  sont assez sensibles au bruit. Le modèle de mixture de Gaussienne offre tout de même de meilleures performances que la sélection statistique : pour 7% de bruit PROB converge dans 80% des cas tandis que GMM converge dans tous les cas. Les relations CMM, BEST, LOOP sont quant-à-elles très peu sensibles au bruit.

**Précision.** (figure 3.10(b)). Elle est mesurée par l'erreur de recalage en pixels lorsque l'algorithme converge. Par rapport au déplacement les différentes relations linéaires par morceaux donnent des résultats similaires, et par conséquent les graphes ne sont pas montrés. La précision obtenue tourne aux alentours de 0,025 pixels quelque soit le déplacement. Les relations GMM et PROB sont moins précises que les autres à partir de 7% de bruits. A 9% Leur précision moyenne est de l'ordre de 0,7 pixels contre 0,3 pixels pour les relations CMM, BEST et LOOP, soit un rapport de 2,5.

**Complexité de convergence.** (figure 3.10(a)). Le complexité de convergence est définie comme le nombre d'itérations nécessaires pour converger. Les relations linéaires par morceaux BEST et LOOP sont clairement les moins efficaces en fonction de l'amplitude du déplacement : LOOP requiert au moins 30 itérations pour converger tandis que BEST n'en nécessite que 10 mais le coût de chacune de ses itérations est plus élevée. CMM, GMM et PROB convergent en moins de 10 itérations quelque soit l'amplitude de déplacement. Notons que les mêmes conclusions sont obtenues lorsque la variance du bruit varie. Les graphes associés ne sont pas montrés.

**Discussion.** Globalement, la relation linéaire par morceaux CMM apparaît avoir les meilleurs performances. Elle est bien plus efficace que les approches bouclées LOOP et BEST et moins sensible au bruit que celles s'appuyant sur des densités de probabilité apprises au préalable : CMM et PROB. Son bassin de convergence n'est que très légèrement moins large que celui des autres relations. Notons que des résultats similaires ont été obtenus avec une transformation de type plaque mince. Dans la suite de ce chapitre nous utilisons la relation linéaire par morceaux CMM lors de nos expérimentations.

### 3.4.2 Comparaison entre les différentes approches existantes

Nous comparons quatre algorithmes en termes de fréquence de convergence, de précision et de complexité de convergence. Deux algorithmes classiques :

- ▷ **DA-GN.** Approximation de Gauss-Newton, mise à jour additive (Bartoli and Zisserman, 2004; Lim and Yang, 2005), voir §2.2.2.3.
- ▷ **DA-ESM.** Approximation du second ordre (Malis, 2004), mise à jour additive, voir §2.2.2.3.

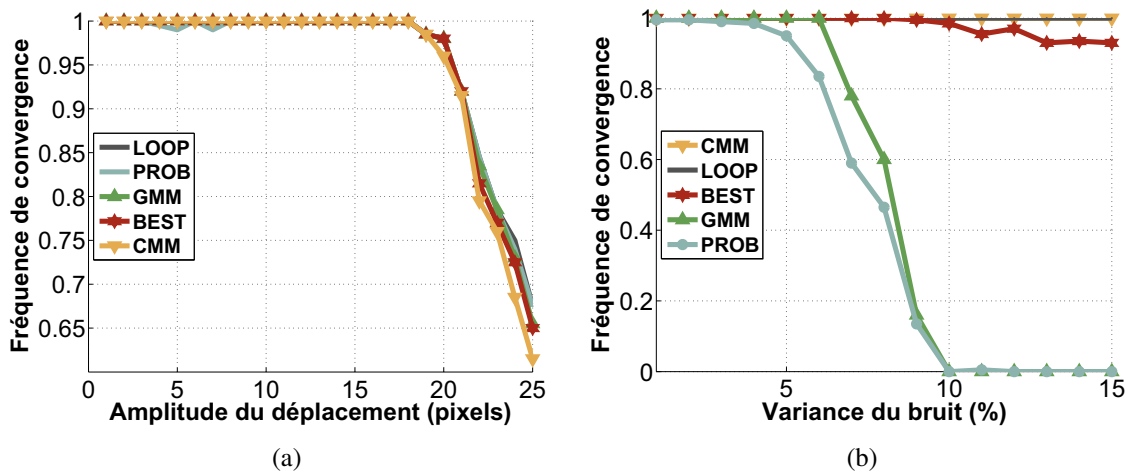


FIG. 3.9 – **Comparaison des différentes relations linéaires par morceaux en terme de fréquence de convergence.** (a) Lorsque l'amplitude de la déformation varie. (b) Lorsque la variance du bruit varie. Pour le graphe (b), les courbes associées à CMM et LOOP sont superposées.

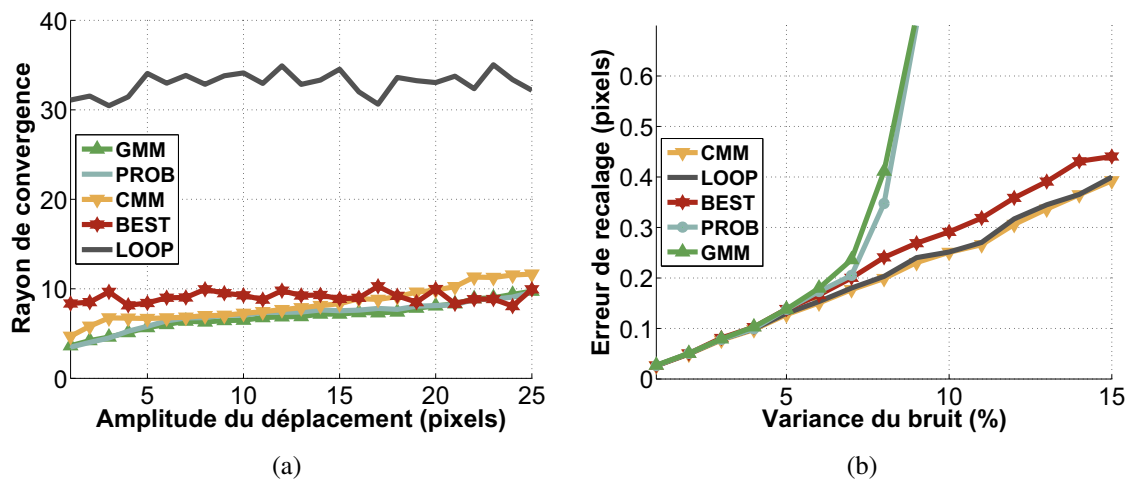


FIG. 3.10 – **Comparaison des différentes relations linéaires par morceaux en terme de complexité de convergence et de précision.** (a) Comparaison des différentes relations linéaires par morceaux en terme de complexité de convergence lorsque l'amplitude de la déformation varie. (b) Comparaison des différentes relations linéaires par morceaux en terme de précision de recalage lorsque la variance du bruit varie.

Deux algorithmes que nous proposons :

- ▷ **IC-GN**. Recalage de type compositionnel inverse guidé par primitives, voir §3.2, avec approximation de Gauss-Newton pour le recalage local.
- ▷ **DC-Le**. Recalage de type compositionnel directe guidé par primitives, voir §3.2, le recalage local étant réalisé par la relation linéaire par morceaux CMM, décrite en §3.3.3.

### 3.4.2.1 Données simulées

Afin d'évaluer ces algorithmes dans des conditions contrôlées, des images synthétiques sont générées à partir de l'image de texture. Les primitives sont placées sur une grille de taille  $3 \times 3$ . Nous utilisons le même protocole qu'en §3.4.1 pour générer des images synthétiques et comparer les différents algorithmes, à l'exception de l'amplitude des déplacements  $r$  dont la valeur par défaut est 3 pixels.

**Fréquence de convergence.** (figure 3.11). DC-Le possède le bassin de convergence le plus large, suivi de très près par DA-ESM tandis que IC-GN a le plus faible bassin de convergence. Pour un déplacement de 8 pixels, DC-Le converge dans environ 75% des cas tandis que DA-GN et IC-GN ne convergent que sur 40% des exemples traités. DA-GN a les moins bonnes performances par rapport au bruit, les autres algorithmes étant très peu sensibles notamment IC-GN qui converge dans tous les cas même pour 15% de bruit.

**Précision.** (figure 3.12). Les quatre algorithmes sont équivalents en terme de précision lorsque l'amplitude de la déformation varie. En ce qui concerne le bruit, IC-GN et DC-Le sont équivalents tandis que DA-ESM est légèrement moins bon et DA-GN clairement moins précis lorsque la variance est grande.

**Complexité de convergence.** (figure 3.13). La complexité de convergence des algorithmes DC-Le et DA-ESM est à peu près constante par rapport à l'amplitude du déplacement et au bruit. Cependant, DC-Le est le meilleur avec une complexité de convergence toujours inférieure à 10. DA-GN et IC-GN sont efficaces pour de petits déplacements c'est-à-dire inférieurs à 5 pixels. Leur complexité de convergence augmente dramatiquement au delà de cette valeur. DA-GN est également inefficace lorsque l'amplitude du bruit est supérieure à 4%. Ceci s'explique par le fait que la matrice Jacobienne de DA-GN dépend majoritairement du gradient de l'image courante, là où le bruit est appliqué.

### 3.4.2.2 Données réelles

Les quatre algorithmes décrits précédemment sont comparés sur plusieurs vidéos. Les informations suivantes sont mesurées : l'erreur résiduelle (RMS<sup>9</sup>) moyenne et maximum en terme d'intensité, au cours de la vidéo, exprimée en UI<sup>10</sup>, le nombre moyen d'itérations ainsi que le temps mis pour traiter la vidéo. Tous les algorithmes ont été implantés en Matlab. Pour pouvoir illustrer la qualité du recalage, nous avons défini un maillage sur l'image de texture qui est transféré sur chaque image de la vidéo. Les différences de recalage entre les quatre algorithmes ne peuvent pas être distinguées visuellement quand ces derniers convergent.

<sup>9</sup>"Root Mean Square".

<sup>10</sup>Unité d'Intensité.

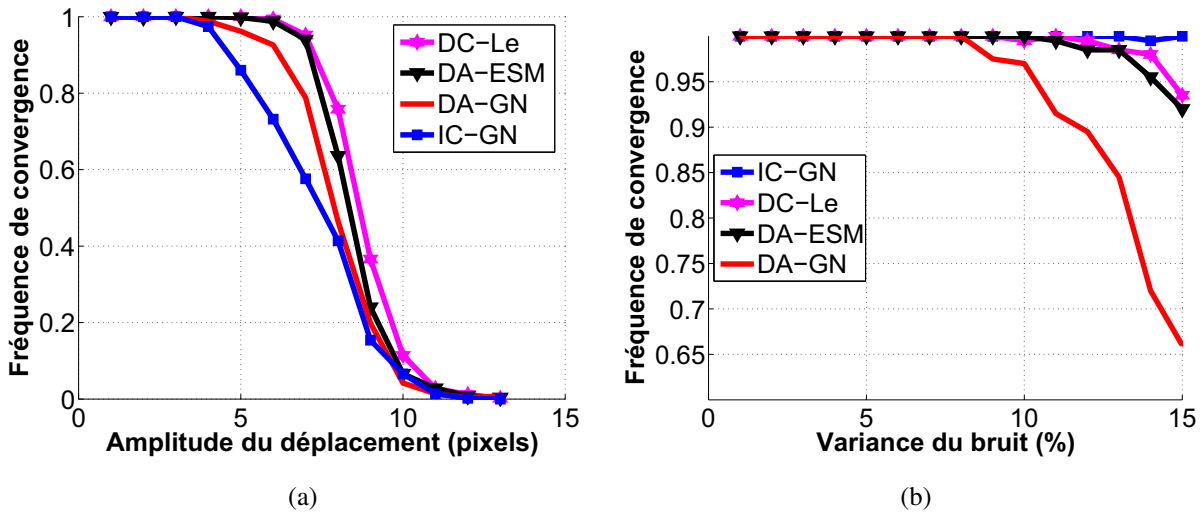


FIG. 3.11 – Comparaison des quatre algorithmes en terme de fréquence de convergence. (a) Lorsque l'amplitude de la déformation varie. (b) Lorsque la variance du bruit varie.

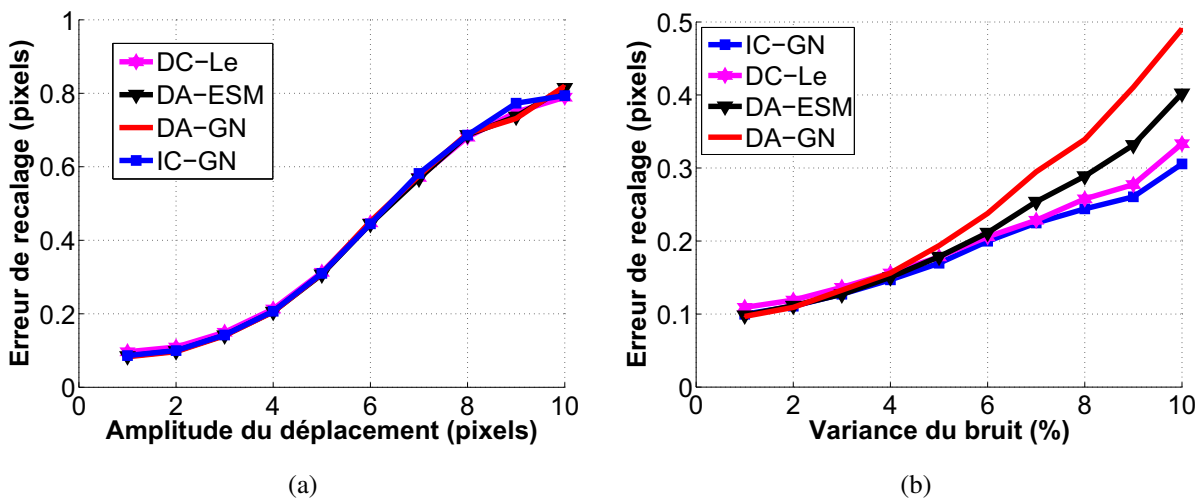


FIG. 3.12 – Comparaison des quatre algorithmes en terme de précision. (a) Lorsque l'amplitude de la déformation varie. (b) Lorsque la variance du bruit varie.

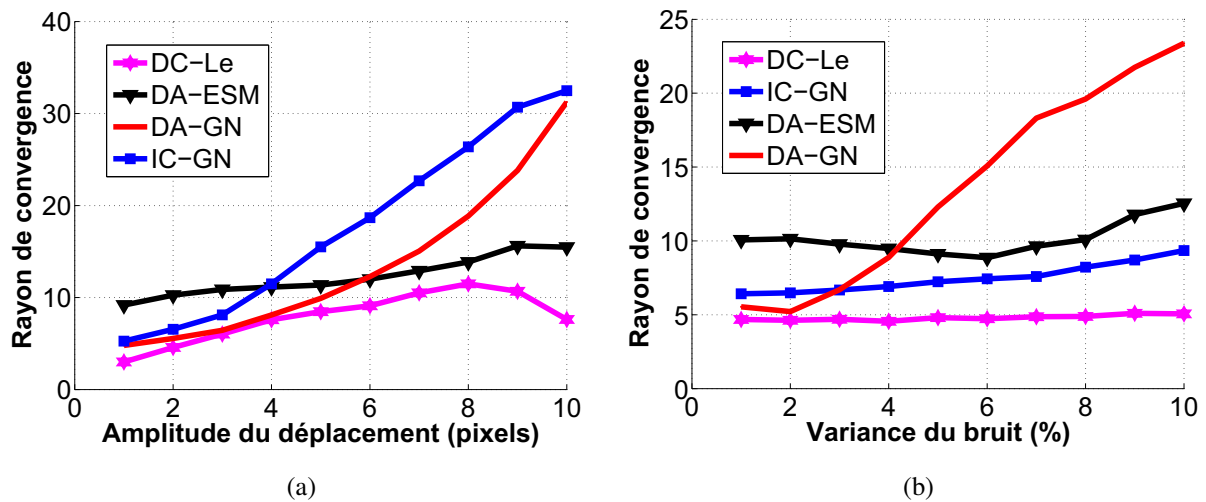


FIG. 3.13 – Comparaison des quatre algorithmes en terme de complexité de convergence. (a) Lorsque l’amplitude de la déformation varie. (b) Lorsque la variance du bruit varie.

**La vidéo du Tshirt.** Cette vidéo est composée de 400 images, l’amplitude des déformations de la surface est parfois importante. Une grille de  $3 \times 3$  primitives est utilisée. Les résultats sont donnés sur le tableau 3.4 et des exemples de déformations estimées par l’algorithme DC-Le sont montrés sur la figure 3.14. DC-Le fonctionne bien sur cette vidéo. En effet, il est à la fois l’algorithme le plus rapide et le plus précis. DA-GN, DA-ESM et IC-GN sont équivalents en terme de précision. Cependant, DA-GN nécessite un grand nombre d’itérations pour converger, le rendant 5 fois plus lent que DC-Le.

	RMS moyen/maximum (UI)	Nombre d’itérations moyen	Temps total/moyen (s)
DA-GN	8,7/13,7	22,7	2.083/5,2
DA-ESM	9,2/14,8	9,2	877/2,2
IC-GN	9,7/15,8	15,6	436/1,1
DC-Le	<b>6,7/12,9</b>	<b>8,3</b>	<b>380/0,9</b>

TAB. 3.4 – Résultat sur la vidéo du Tshirt. Les meilleurs résultats sont mis en gras.

**La vidéo de la feuille de papier.** Cette vidéo est composée de 350 images. Les résultats sont donnés sur le tableau 3.5. Une grille de  $4 \times 4$  primitives est utilisée. IC-GN diverge lorsque les déformations semblent être les plus importantes. Les autres algorithmes ont des performances sensiblement similaires. DA-GN est tout de même le plus précis, et DC-Le est 3 fois plus rapide. La figure 3.15 illustre les déformations estimées par l’algorithme DC-Le ainsi que l’augmentation 2D de la vidéo de la feuille de papier. Nous avons également capturé les déformations de la surface observée en estimant, au sens des moindres carrés, les transformations de similarité entre l’image de texture et chaque image de la vidéo. Une fois les mouvements globaux compensés, seules les déformations sont appliquées sur une nouvelle texture.

**La vidéo du tapis.** Cette courte vidéo est composée de 42 images, l’amplitude des déformations est assez importante, elles sont réparties sur toute la surface. Une grille de  $5 \times 5$  primitives est utilisée.

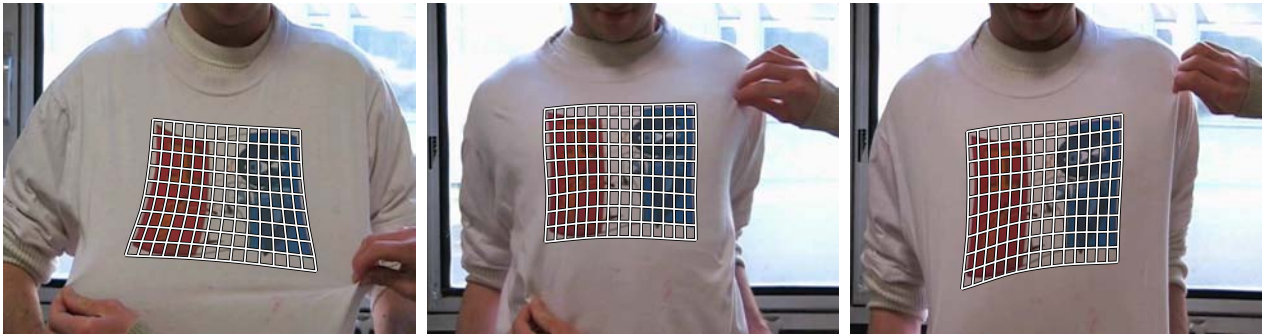


FIG. 3.14 – Déformations estimées par l’algorithme DC-Le sur la vidéo du Tshirt.

	RMS moyen/maximum (UI)	Nombre d’itérations moyen	Temps total/moyen (s)
DA-GN	<b>9,0/17,6</b>	6,9	532/1,5
DA-ESM	10,2/20,5	7,1	560/1,6
DC-Le	9,4/19,4	<b>3,8</b>	<b>176/0,5</b>

TAB. 3.5 – **Résultat sur la vidéo de la feuille de papier.** IC-GN diverge sur cette vidéo. Les meilleurs résultats sont mis en gras.

Les performances des différents algorithmes sont répertoriées sur le tableau 3.6, les déformations estimées par l’algorithme DC-Le sont illustrées sur la figure 3.16. Comme pour la vidéo de la feuille de papier, l’algorithme IC-GN diverge. DA-GN et DA-ESM recalent précisément les images. DC-Le est légèrement moins précis mais il est 7 fois plus rapide.

	RMS moyen/maximum (UI)	Nombre d’itérations moyen	Temps total/moyen (s)
DA-GN	<b>5,6/8,2</b>	12,8	118/2,8
DA-ESM	5,8/8,6	11,4	109/2,6
DC-Le	6,5/9,8	<b>3,5</b>	<b>17,1/0,4</b>

TAB. 3.6 – **Résultat sur la vidéo du tapis.** IC-GN diverge sur cette vidéo. Les meilleurs résultats sont mis en gras.

**La deuxième vidéo du Tshirt.** Cette vidéo est composée de 623 images, l’amplitude des déformations y est modérée. Une grille de  $3 \times 3$  primitives est utilisée. Des variations globales d’illumination apparaissent au cours de la vidéo. Une transformation photométrique de type gain et biais est explicitement incorporée au processus d’optimisation. Celle-ci est évaluée pour chaque image en calculant la moyenne et la variance des intensités. Les performances des différents algorithmes sont répertoriées sur le tableau 3.6. Les déformations estimées par l’algorithme DC-Le sont illustrées sur la figure 3.16. Les différents algorithmes offrent de bonnes performances sur cette vidéo, les erreurs résiduelles faibles prouvent que les variations globales d’illumination sont correctement compensées. DC-Le et IC-GN sont respectivement 4 et 2 fois plus rapides par rapport aux approches classiques à mise à jour additive.



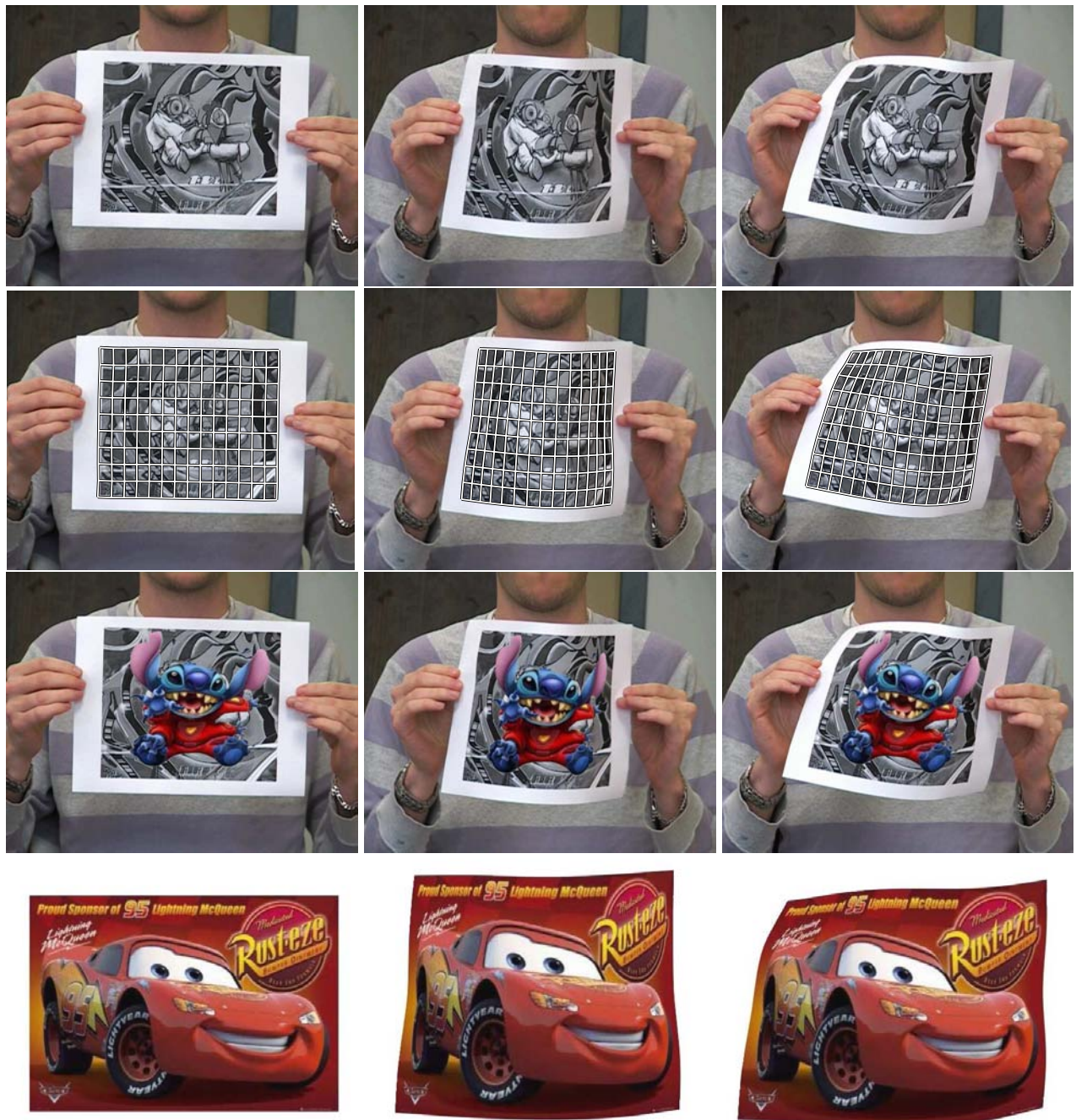


FIG. 3.15 – Résultat du recalage et de l'augmentation 2D pour l'algorithme DC-Le sur la vidéo de la feuille de papier. 2<sup>ème</sup> ligne : déformations estimées par l'algorithme DC-Le. 3<sup>ème</sup> ligne : augmentation 2D de la vidéo du papier avec un logo du personnage *Stitch*. En bas : les déformations sont capturées et appliquées sur une affiche du film *Cars*.

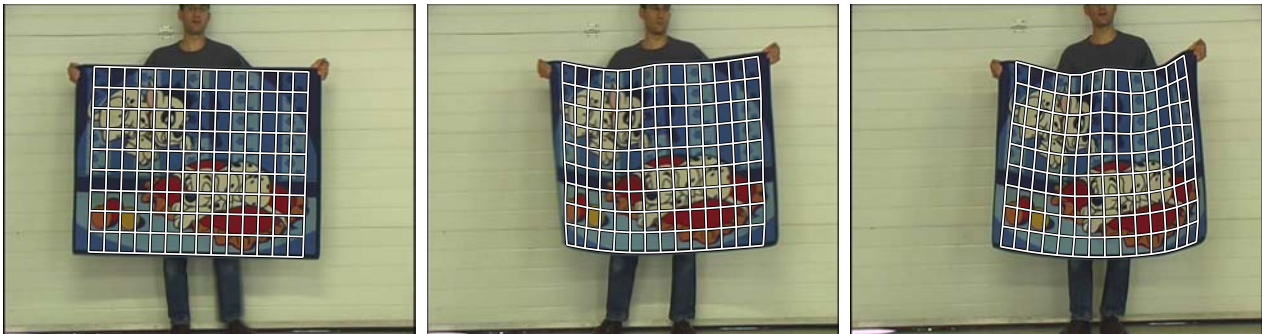


FIG. 3.16 – Déformations estimées par l’algorithme DC-Le sur la vidéo du tapis.

	RMS moyen/maximum (UI)	nombre d’itérations	Temps total/moyen
DA-GN	<b>4,5/7,4</b>	5,5	785/1,3
DA-ESM	4,5/7,5	5,2	788/1,3
IC-GN	4,9/7,6	7,1	381/0,6
DC-Le	4,6/7,7	<b>2,8</b>	<b>247/0,4</b>

TAB. 3.7 – **Résultat sur la deuxième vidéo du Tshirt.** Les meilleurs résultats sont mis en gras.



FIG. 3.17 – Déformations estimées par l’algorithme DC-Le sur la deuxième vidéo du Tshirt.

**Discussion.** Il a été remarqué sur un grand nombre de données réelles que DA-GN reste l'algorithme le plus précis. Cependant, il n'est pas efficace en terme de temps de calcul et ceci est d'autant plus vrai que le déplacement est important. DA-ESM présente des performances de recalage légèrement supérieure à celle de DA-GN ; il est notamment un peu plus efficace. IC-GN est un algorithme efficace mais perd en précision pour des déplacements très importants et peut diverger. De façon générale, DC-Le présente le meilleur compromis précision / vitesse. En effet, il se trouve être quasiment aussi précis que DA-GN tout en étant 5 fois plus rapide en moyenne. De plus, il est équivalent à, ou meilleur que IC-GN et DA-ESM en terme de précision de recalage et de temps de calcul, et possède un bassin de convergence plus large.

## 3.5 Conclusion

Deux contributions importantes ont été présentées dans ce chapitre pour le problème du recalage d'images d'une surface déformable. Tout d'abord, nous proposons de guider les fonctions de déformation par des primitives, permettant de relâcher la contrainte de groupe et d'utiliser les algorithmes à mise à jour compositionnelle plus efficaces que les méthodes additives habituellement privilégiées. Ensuite, nous proposons une relation linéaire par morceaux pour l'étape de recalage local des algorithmes basés apprentissage. Cette dernière présente de bien meilleures performances qu'une simple approximation linéaire si les différentes matrices d'interaction apprises sont combinées de manière optimale. L'association de ces deux contributions introduit un algorithme de recalage d'images dont les performances en terme de précision, de temps de calcul et de fréquence de convergence dépassent ou égalent celles des algorithmes existants.

En perspective, la sélection automatique de la complexité de la fonction de déformation devra être étudiée. Pour l'heure, le nombre de primitives utilisées est fixé de manière empirique. Une solution existante est l'insertion dynamique de centres (Bartoli and Zisserman, 2004). Cependant, elle entraîne une perte d'efficacité de IC-GN puisque la matrice Jacobienne n'est alors plus constante, pour DC-Le un apprentissage pour toutes les combinaisons possibles position - nombre de centres est alors requis. Ceci n'est pas raisonnablement réalisable en pratique.

Le nombre de centres de la fonction de déformation peut être fixe quelle que soit la vidéo à traiter à condition qu'ils soient suffisamment nombreux et repartis de manière homogène pour garantir la flexibilité de la fonction de déformation. En contrepartie, elle doit être fortement régularisée afin d'être suffisamment contrainte. La sélection automatique du paramètre contrôlant l'influence de la régularisation fait l'objet de nombreux travaux de recherche. Des méthodes efficaces existent (Bartoli, 2007) pour lesquelles un critère basé primitives est requis. Leur extension à un critère direct nécessite des études complémentaires.

L'algorithme DC-Le ne gère pas sous sa forme d'origine la présence d'occultations et de variations locales d'illumination. L'emploi de « patches » est une solution possible (Shi and Tomasi, 1994). Dans le cadre des surfaces déformables, la question du choix de la complexité de la transformation associée à un patch se pose : utiliser une fonction de déformation pour modéliser le mouvement d'un petit patch n'a pas réellement de sens. Une autre problématique est la combinaison des informations issues des différents patches pour aboutir à la fonction de déformation globale.



# Chapitre 4

## Gestion des auto-occultations en recalage d'images

---

---

Dans ce chapitre, nous adressons le problème du recalage d'images d'une surface déformable en présence d'auto-occultations. Un cadre spécifique permettant de gérer ces dernières est proposé. L'idée principale est de forcer la fonction de déformation à se comprimer dans les régions auto-occultées. Celles-ci sont alors détectées grâce à cette propriété de contraction. Des résultats expérimentaux sur des vidéos difficiles montrent que notre approche recalcule précisément les images tout en détectant les régions auto-occultées. Ce cadre de gestion des auto-occultations est utilisé pour des applications d'augmentation 2D de surfaces déformables sur une vidéo. La méthodologie permettant de gérer les auto-occultations est publiée dans (Gay-Bellile et al., 2007b, 2008) et les différentes améliorations requises pour l'augmentation 2D sont exposées dans (Gay-Bellile et al., 2007d,c). L'ensemble des contributions présentées ci-dessous sont récapitulées dans un article de revue (Gay-Bellile et al., Soumis en 2008). Nous sommes en attente de la décision finale après avoir procédé à une première révision de l'article.

---

---

### 4.1 Introduction et travaux antérieurs

Les occultations créent des problèmes majeurs en recalage d'images. Elles rendent celui-ci beaucoup plus complexe car elles introduisent des variations d'apparence importantes et des observations erronées. Elles peuvent être de deux types : les *occultations externes* et les *auto-occultations*. Les premières apparaissent lorsqu'un objet passe entre la caméra et la surface observée. Les auto-occultations quant à elles surviennent lorsque la surface est pliée de telle manière qu'une partie en cache une autre. Nous nous intéressons plus particulièrement à ces dernières car les auto-occultations externes sont plutôt bien gérées en recalage d'images (Ladikos et al., 2007; Pilet et al., 2008).

L'estimation des déformations en présence d'auto-occultations est une problématique pouvant être modélisée de manière tridimensionnelle. La détection des auto-occultations se fait alors par des techniques comme le *z-buffer* ou comme la solution d'une équation différentielle ordinaire, lorsque la surface est représentée sous une forme implicite (Ilic et al., 2007). Cependant, évaluer la surface 3D, ses déformations et le mouvement des caméras à partir d'une vidéo prise par une seule caméra est plus

« non-linéaire » que l'estimation d'un champ de déplacement au niveau image. L'information 3D n'est en outre pas nécessaire pour de nombreuses applications, comme par exemple l'augmentation 2D d'images d'une surface déformable. Un modèle de déformation 2D reste fréquemment privilégié pour le recalage d'images. Les travaux précédents utilisant ces modèles, par exemple (Pilet et al., 2008), incluent un M-estimateur dans la fonction de coût afin de rendre leur approche robuste aux données aberrantes, comme par exemple celles introduites par la présence d'auto-occultations et d'occultations externes. La régularisation est privilégiée dans les régions détectées par l'estimateur robuste comme erronées en terme d'intensité lumineuse (ou signature de primitives). Ce comportement est bien adapté pour les occultations externes mais beaucoup moins pour les auto-occultations, comme expliqué plus en détails en §4.2. Les auto-occultations sont explicitement modélisées dans (Lin and Liu, 2006) pour des surfaces présentant une texture répétitive appelée NRT pour "Near Regular Texture". Un élément de base de ces textures est appelé "texton". Leur approche s'appuie sur l'estimation d'une carte de visibilité construite en se basant sur les propriétés répétitives de forme et d'apparence relatives aux NRT. Cette carte prend notamment en compte le comportement de la fonction de déformation en s'appuyant sur les différences de topologie (surface, longueur des côtés) entre textons voisins. Leur algorithme permet de suivre les déformations d'une surface en dépit d'auto-occultations assez importantes.

Nous avons constaté que très peu de travaux prétendent gérer les auto-occultations. Les différents résultats présentés dans la littérature en attestent car les expériences où la surface s'auto-occulte significativement sont assez rares.

**Contributions.** Nous proposons un cadre spécifique de gestion des auto-occultations modélisant explicitement ces dernières :

- ▷ **Contraction de la fonction de déformation.** Nous imposons à la fonction de déformation de se comprimer le long de la frontière d'auto-occultation. Pour cela, un troisième terme appelé le *contracteur*, présenté en §2.2.1, est ajouté à la fonction de coût classique.
- ▷ **Détection des pixels auto-occultés.** La propriété de contraction permet d'utiliser le comportement de la fonction de déformation pour détecter les régions auto-occultées. Notre opérateur de détection se base sur le principe que la contraction introduit une dérivée nulle de la fonction de déformation dans au moins une direction.

L'algorithme résultant ne requiert aucune texture spécifique et est général puisque la seule hypothèse est que les déformations de la surface observée soient lisses. Cette hypothèse est valide pour de nombreux scénarios. Les résultats expérimentaux montrent que l'approche proposée est bien mieux adaptée aux auto-occultations que les méthodes robustes classiques ne modélisant pas explicitement ces dernières. En effet, elle permet de recalibrer précisément les régions apparentes, ce qui est indispensable pour pouvoir réestimer correctement la fonction de déformation lorsque l'auto-occultation cesse. Les régions non visibles sont quant à elles transférées sur la frontière d'auto-occultation. Notre approche est testée dans le cadre d'applications de suivi et d'augmentation 2D de surfaces déformables pouvant s'auto-occulter.

**Organisation du chapitre.** Les problèmes inhérents aux auto-occultations ainsi que la solution proposée sont énoncés en §4.2. La détection des auto-occultations est étudiée en §4.3. En §4.4, nous décrivons l'algorithme permettant de recalibrer des images d'une surface déformable auto-occultée. L'augmentation 2D de surfaces déformables sur une vidéo est traité en §4.5. Les résultats expérimentaux

sont présentés en §4.6. Pour finir, nous donnons nos conclusions et discutons des travaux futurs en §4.7.

## 4.2 Énoncé du problème et solution proposée

### 4.2.1 Problématique

Les algorithmes robustes de recalage d'images décrits au chapitre 2 sont bien adaptés pour les occultations externes. En effet, le terme de régularisation devient prépondérant pour les pixels détectés comme occultés. La fonction de déformation reste lisse au niveau de l'occultation, ce qui est le comportement souhaité. La détection des régions occultées s'appuie sur le terme de données  $\mathcal{E}_d$ ; une différence d'intensité pour les approches directes. Elle distingue par simple seuillage de l'image de différence l'occultant de l'occulté, sous l'hypothèse qu'ils soient sensiblement différents en terme d'intensité lumineuse.

En cas d'auto-occultation, l'utilisation de ce critère s'avère inefficace. Les auto-occultations requièrent une grande précision de la détection puisqu'elles sont introduites par de fortes déformations. La région d'intérêt étant en outre souvent homogène en terme d'intensité, il devient plus difficile de lever l'ambiguïté entre les régions visibles et non-visibles, en se basant uniquement sur l'information contenue dans les images. Des erreurs de recalage apparaissent inévitablement avec ce module de détection.

### 4.2.2 Approche proposée

Nous proposons d'utiliser le comportement de la fonction de déformation le long de la frontière d'auto-occultation pour détecter les pixels auto-occultés. Il peut être de deux types pour des fonctions continues, sous l'hypothèse que les régions visibles soient correctement recalées : soit la fonction de déformation est rabattue sur elle même (boucle), soit elle est comprimée au niveau de la frontière d'auto-occultation. La première possibilité représente le comportement naturel d'une fonction de déformation régularisée tandis que la deuxième doit être imposée.

Seule la contraction permet une détection précise des régions auto-occultées en utilisant les dérivées directionnelles de la fonction de déformation, comme nous le décrivons en §4.3.2. Si la fonction de déformation forme une « boucle », alors ses dérivées permettent de déceler la présence de régions auto-occultées mais ne permettent pas de définir précisément les régions associées comme l'illustre la figure 4.1. Cette figure représente une coupe de la fonction de déformation dans la direction perpendiculaire à la frontière d'auto-occultation. Cette coupe est représentée pour les deux configurations possibles décrites ci-dessus : contraction et boucle. Sur cet exemple, lorsque la fonction de déformation se rabat sur elle même on constate une variation du signe de la dérivée dans la direction  $x$  permettant de déceler la présence d'auto-occultation. Cependant, il est impossible de situer précisément les bornes de la région auto-occultée (point vert (ou gris clair) et rouge (ou gris foncé)) le long de la courbe. Par contre, si la fonction de déformation est comprimée le long de la frontière d'auto-occultation alors la région auto-occultée correspond parfaitement à une région où la fonction de déformation possède une dérivée nulle suivant la direction  $x$  : un critère permettant la détection précise des régions auto-occultées peut alors être défini. En résumé, la présence de boucles lors du recalage doit être évitée et remplacée par la contraction de la fonction de déformation.

Pour cela un troisième terme, appelé le contracteur, est ajouté à la fonction de coût classique (équation (2.1) de la section 2.2.1). Il détecte les boucles comme des variations de signe dans les dérivées directionnelles de la fonction de déformation et impose à la place l'annulation des dérivées dans une direction c'est-à-dire la contraction de la fonction de déformation suivant cette direction. Plus de détails sur le contracteur utilisé dans notre algorithme de recalage sont donnés en §4.3.1. Notons que cette représentation possède également l'avantage de faire disparaître naturellement les régions auto-occultées, ce qui est très utile pour des applications d'augmentation 2D d'images d'une surface déformable auto-occultée. Plus de détails sont donnés en §4.5.

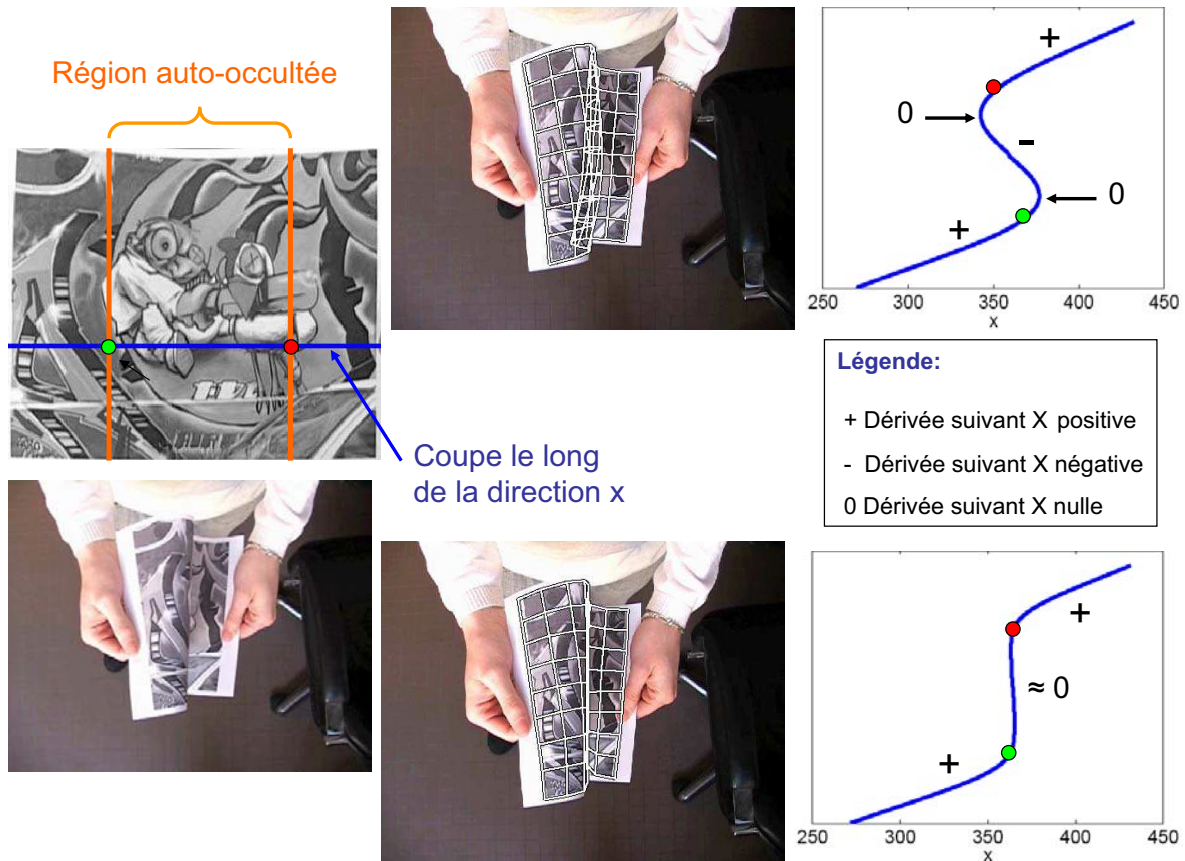


FIG. 4.1 – Coupes de la fonction de déformation pour différentes configurations. A gauche : l'image de référence sur laquelle est surimposée la coupe appliquée à la fonction de déformation ainsi que les bornes définissant la région auto-occultée le long de cette coupe. Au centre : les différentes configurations possibles de la fonction de déformation en présence d'auto-occultation. En haut à droite : évolution de la composante  $x$  de la fonction de déformation le long de la coupe lorsque celle-ci forme une boucle. En bas à droite : évolution de la composante  $x$  de la fonction de déformation le long de la coupe lorsque celle-ci est comprimée.

La détection des régions auto-occultées se fait par un module s'appuyant sur la recherche des dérivées directionnelles faibles de la fonction de déformation. Nous allons voir en §4.3.2 que cette recherche exhaustive peut être remplacée par un problème de minimisation pour chaque pixel dont la solution possède une forme analytique simple.



La figure 4.2 illustre les différences entre la fonction de coût robuste classique et celle proposée, modélisant explicitement les auto-occultations.

### Approche robuste

Termes de la fonction de coût	données	régularisation
Un pixel visible	●	●
Un pixel occulté	●	●

### Notre approche

Termes de la fonction de coût	données	régularisation	contracteur
Un pixel visible	●	●	●
Un pixel occulté par un objet externe	●	●	●
Un pixel auto-occulté	●	●	●

FIG. 4.2 – **Fonction de coût classique et celle proposée modélisant explicitement les auto-occultations.** Les termes qui sont pris en compte dans la minimisation sont représentés par un cercle vert (gris clair) et ceux qui ne sont pas pris en compte sont représentés par un cercle rouge (gris foncé). L'approche classique traite les pixels auto-occultés et ceux occultés par un objet externe de la même manière, c'est-à-dire comme des données aberrantes. Notre approche consiste à activer le contracteur au niveau des pixels auto-occultés ; la détection des régions associées se faisant par le module de détection décrit en §4.3.2.

### 4.2.3 Approche alternative

Récemment, nos travaux ont été repris dans (Hilsmann and Eisert, 2008) pour proposer un nouvel algorithme de recalage d'images d'une surface déformable auto-occultée. L'idée de base est identique au cadre de gestion des auto-occultations que nous proposons. La fonction de déformation est forcée à se comprimer au niveau de la frontière d'auto-occultation. La détection des pixels auto-occultés s'appuie alors sur cette propriété. Cependant les auteurs procèdent différemment : la détection des pixels auto-occultés se fait en se basant sur un critère de rapport de distances évaluées entre un sommet du maillage définissant la fonction de déformation et ses voisins. Ce critère est ensuite interpolé afin d'obtenir une carte d'auto-occultation pixélique. La contraction de la fonction de déformation est quant à elle imposée en pondérant localement le terme de régularisation en fonction du résultat de la détection, évitant ainsi l'ajout d'un troisième terme dans la fonction de coût.

### 4.3 Outils proposés pour la détection des auto-occultations

L'approche proposée pour la détection des pixels auto-occultés retourne deux versions de la carte d'auto-occultation : une carte binaire  $\mathcal{H}(\mathbf{q}; \mathbf{u})$  et une carte probabiliste  $\widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u})$ . La première présente plusieurs inconvénients que nous décrivons ci-dessous. Par conséquent la carte probabiliste est utilisée dans la plupart de nos expériences.

#### 4.3.1 Le contracteur

Le contracteur a pour but d'empêcher la fonction de déformation de se rabattre sur elle-même au niveau de la frontière d'auto-occultation. Ces configurations apparaissent naturellement lorsque la surface s'auto-occulte. Elles sont caractérisées par une variation du signe de la dérivée de la fonction de déformation le long d'une certaine direction : la direction normale à la frontière d'auto-occultation. Le contracteur doit donc pénaliser les variations de signe de la dérivée de la fonction de déformation et imposer à la place la contraction, c'est-à-dire une dérivée nulle dans cette direction.

Il est construit à partir de la fonction  $\gamma$  représentée sur la figure 4.3(a) et définie par :

$$\gamma(a) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{si } a \geq 0 \\ a^2 & \text{sinon.} \end{cases}$$

Cette fonction est appliquée au produit, élément par élément, entre les dérivées directionnelles à gauche et à droite de la fonction de déformation évaluées en tout point  $\mathbf{q} \in \mathbb{R}$  et intégrées sur les directions  $\mathbf{d} \in \mathbb{S}^1$ .

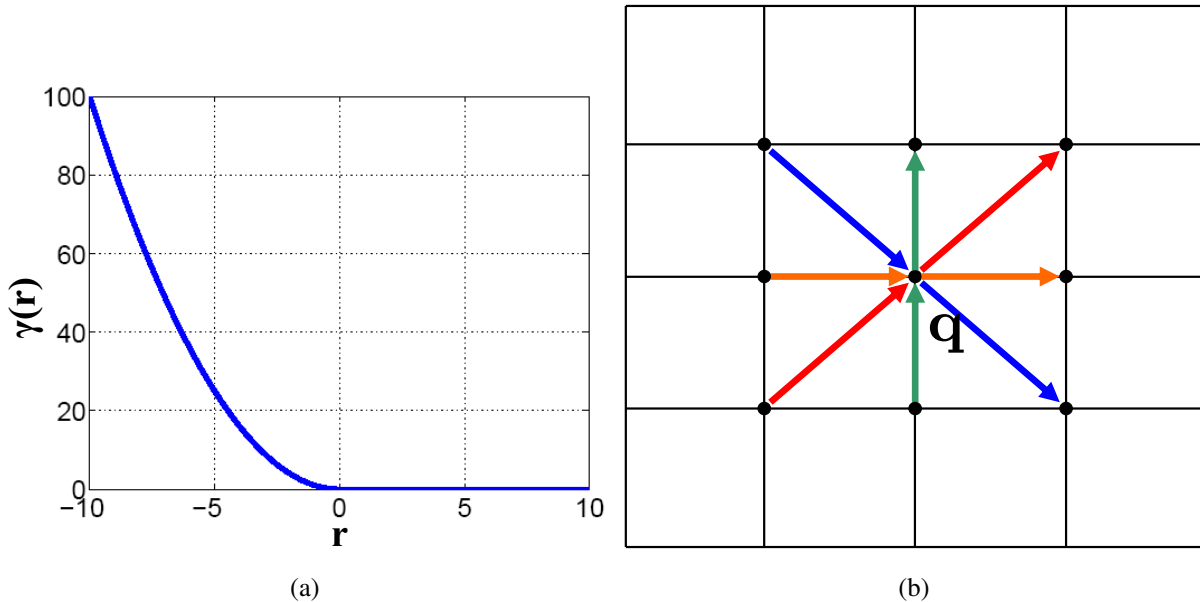


FIG. 4.3 – Les composants du contracteur. (a) La fonction  $\gamma$ . (b) L'ensemble  $\mathcal{F}$  des directions discrétisées.

Le contracteur est donné par :

$$\mathcal{E}_f(\mathbf{u}) \stackrel{\text{def}}{=} \sum_{\mathbf{q} \in \mathcal{R}} \int_{\mathbf{d} \in \mathbb{S}^1} \sum_{p \in \{x, y\}} \gamma(\mathbf{E}_l^p(\mathbf{d}; \mathbf{q}; \mathbf{u}) \mathbf{E}_r^p(\mathbf{d}; \mathbf{q}; \mathbf{u})) d\mathbf{d}.$$

La minimisation de ce terme fait que les contraintes décrites ci-dessus sont respectées. Tout d'abord, seuls les points pour lesquels les dérivées à droite et à gauche sont de signes opposés sont pénalisés puisque la fonction  $\gamma$  est nulle pour tout argument positif. Ensuite, le minimum de  $\mathcal{E}_f(\mathbf{u})$  est obtenu lorsque la fonction de déformation ne se rabat pas.

En pratique, l'intégrale est discrétisée sur un ensemble  $\mathcal{F}$  de directions représentées sur la figure 4.3(b). Le contracteur devient :

$$\mathcal{E}_f(\mathbf{u}) \approx \sum_{\mathbf{q} \in \mathcal{R}} \sum_{\mathbf{d} \in \mathcal{F}} \sum_{p \in \{x,y\}} \gamma(\mathbf{E}_l^p(\mathbf{d}; \mathbf{q}; \mathbf{u}) \mathbf{E}_r^p(\mathbf{d}; \mathbf{q}; \mathbf{u})). \quad (4.1)$$

## 4.3.2 Cartes d'auto-occultation binaires

Deux modules de détection d'auto-occultations sont décrits ci-dessous. Ils introduisent des cartes d'auto-occultation binaires.

### 4.3.2.1 Approche basée sur la position

La fonction de déformation est contrainte à se comprimer le long de la *frontière d'auto-occultation*. Les pixels de l'image de référence transférés sur les mêmes positions dans l'image courante doivent être étiquetés comme auto-occultés. Soit un pixel  $\mathbf{q} \in \mathcal{R}$ , il est considéré comme auto-occulté si il existe au moins un autre pixel  $\mathbf{q}' \in \mathcal{R}$  tel que la distance entre  $\mathbf{q}$  et  $\mathbf{q}'$  soit supérieure à un seuil  $r_t$  et que leur distance, après avoir été transférés sur l'image courante par  $\mathcal{W}$ , soit inférieure à un autre seuil  $r_i$ . La figure 4.4 illustre ce principe. Notons que  $r_t$  et  $r_i$  sont choisis tel que  $r_t > r_i$ . La carte d'auto-occultation introduite par le *module basé position* est donnée par :

$$\mathcal{H}_t(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \begin{cases} 1 & \exists \mathbf{q}' \in \mathcal{R} \mid \|\mathbf{q} - \mathbf{q}'\| > r_t \text{ et } \|\mathcal{W}(\mathbf{q}; \mathbf{u}) - \mathcal{W}(\mathbf{q}'; \mathbf{u})\| < r_i \\ 0 & \text{sinon.} \end{cases}$$

Les valeurs typiques des seuils sont par exemple  $r_t = 1$  et  $r_i = 0,1$ . Cette définition mathématique ne peut pas directement être utilisée en pratique puisque pour chaque pixel  $\mathbf{q} \in \mathcal{R}$ , tous les pixels de  $\mathcal{R}$  doivent être testés. La complexité du module de détection est alors quadratique sur le nombre de pixels.

Cependant, elle peut être réduite en construisant une liste  $\mathcal{L}_{\mathbf{q}_i}$  pour chaque pixel  $\mathbf{q}_i$  de l'image courante. Un pixel  $\mathbf{q}$  est contenu dans  $\mathcal{L}_{\mathbf{q}_i}$  si  $\mathbf{q}_i$  est le pixel de l'image cible le plus proche de  $\mathcal{W}(\mathbf{q}; \mathbf{u})$ . La condition  $\|\mathcal{W}(\mathbf{q}; \mathbf{u}) - \mathcal{W}(\mathbf{q}'; \mathbf{u})\| < r_i$  ne peut être respectée que si les pixels  $\mathbf{q}$  et  $\mathbf{q}'$  appartiennent à des listes de pixels voisins dans l'image courante. Pour pouvoir détecter si un pixel  $\mathbf{q} \in \mathcal{L}_{\mathbf{q}_i}$  est auto-occulté, la recherche se limite alors aux pixels appartenant également à la liste de  $\mathbf{q}_i$  et à celles de son voisinage, réduisant ainsi la complexité du problème.

### 4.3.2.2 Approche basée sur la dérivée

Si la fonction de déformation se comprime au point  $\mathbf{q}$  dans la direction  $\mathbf{d}$ , alors les points  $\mathbf{q} + \epsilon \mathbf{d}$  et  $\mathbf{q} - \epsilon \mathbf{d}$  sont transférés par  $\mathcal{W}$  sur la même position :  $\mathcal{W}(\mathbf{q} + \epsilon \mathbf{d}; \mathbf{u}) \approx \mathcal{W}(\mathbf{q} - \epsilon \mathbf{d}; \mathbf{u})$  comme l'illustre la figure 4.5. Ceci implique que la dérivée directionnelle dans la direction  $\mathbf{d}$  évaluée au point  $\mathbf{q}$  est approximativement nulle :  $\frac{\partial_a \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q}; \mathbf{u}) \approx 0$ . Une conséquence liée à la contraction de la fonction de déformation est qu'il existe, pour chaque pixel  $\mathbf{q}$  auto-occulté, une direction  $\mathbf{d} \in \mathbb{S}^1$  pour laquelle sa dérivée directionnelle s'annule.

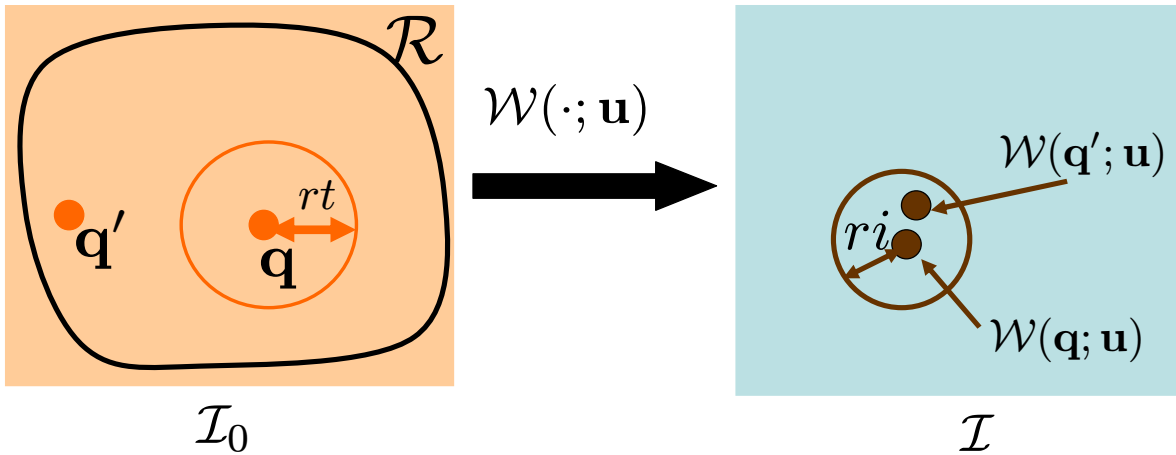


FIG. 4.4 – **Principe du module de détection des auto-occultations basé position.** Deux pixels  $\mathbf{q}$  et  $\mathbf{q}'$  sont détectés comme auto-occultés si leur distance après avoir été transférés sur l'image courante  $\mathcal{I}$  par  $\mathcal{W}$  est inférieure au seuil  $r_i$ , sous la condition que leur distance d'origine soit supérieure au seuil  $r_t$ . Dans cet exemple, les pixels  $\mathbf{q}$  et  $\mathbf{q}'$  sont détectés comme auto-occultés.

La carte d'auto-occultation introduite par le *module basé dérivé* est donnée par :

$$\mathcal{H}_d(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \begin{cases} 1 & \exists \mathbf{d} \in \mathbb{S}^1 \mid \|\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2 < r_d \\ 0 & \text{sinon,} \end{cases} \quad (4.2)$$

où  $\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})$  est l'approximation par différence finie de  $\frac{\partial \mathcal{W}}{\partial \mathbf{q}}(\mathbf{q}; \mathbf{u})$ . On fixe la valeur de  $r_d$  légèrement supérieure à 0 afin de tolérer du bruit sur la fonction de déformation et d'anticiper les auto-occultations, c'est-à-dire de les détecter juste avant qu'elles n'apparaissent réellement.

En pratique, la recherche exhaustive des directions  $\mathbf{d}$  imposée par l'équation (4.2) est remplacée par un problème de minimisation :

$$\mathcal{H}_d(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \left( \left( \min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2 \right) < r_d \right).$$

Le test de comparaison (ici inférieur) renvoie des valeurs booléennes « vrai » ou « faux » converties en valeurs binaires 1 ou 0.

Ce problème possède une solution analytique simple. Soit  $\mathbf{E}(\mathbf{q}; \mathbf{u})$  la matrice Jacobienne de  $\mathcal{W}$  évaluée à  $(\mathbf{q}; \mathbf{u})$  par différence finie, on obtient  $\mathbf{E}_c(\mathbf{d}; \mathbf{q}; \mathbf{u}) \approx \mathbf{E}(\mathbf{q}; \mathbf{u})\mathbf{d}$ , et par suite :

$$\sigma_0 = \min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}_c(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2 \approx \min_{\mathbf{d} \in \mathbb{S}^1} \mathbf{d}^\top \mathbf{E}(\mathbf{q}; \mathbf{u})^\top \mathbf{E}(\mathbf{q}; \mathbf{u}) \mathbf{d}.$$

La solution est donnée par la plus petite valeur propre de  $\mathcal{O} \stackrel{\text{def}}{=} \mathbf{E}(\mathbf{q}; \mathbf{u})^\top \mathbf{E}(\mathbf{q}; \mathbf{u})$ . Une décomposition spectrale de la matrice  $\mathcal{O}$ , pour laquelle il existe une solution analytique, donne :

$$\sigma_0 = \frac{1}{2} \left( O_{1,1} + O_{2,2} - \sqrt{(O_{1,1} - O_{2,2})^2 + 4O_{1,2}^2} \right).$$

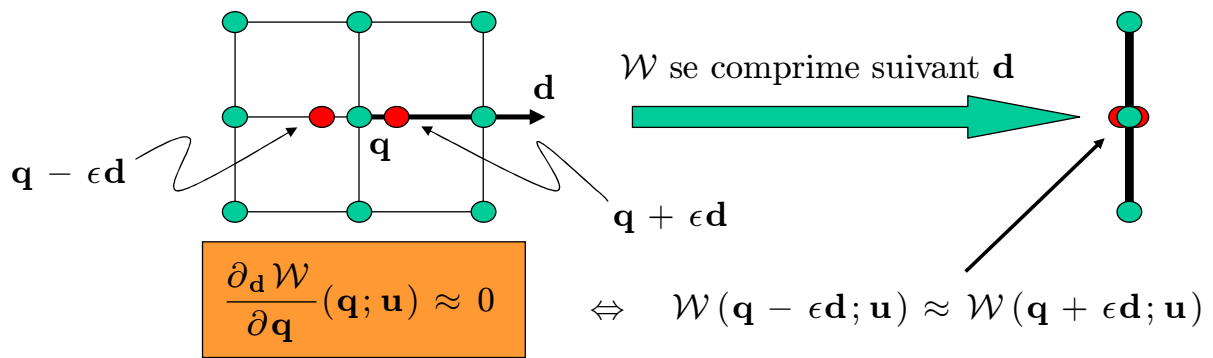


FIG. 4.5 – **Principe du module de détection des auto-occultations basé dérivée.** Une contraction de la fonction de déformation suivant la direction  $\mathbf{d}$  entraîne une dérivée directionnelle nulle dans cette direction. La recherche de dérivées directionnelles nulles (ou proches de zéro) permet de déceler la présence de régions auto-occultées.

#### 4.3.2.3 Discussion sur la méthode de détection

Le module de détection basé dérivée est utilisé au cours de nos expérimentations. Il dépend d'un seul seuil dont la valeur peut être fixée intuitivement, contrairement au module basé sur la position, qui dépend de deux seuils. Notons cependant que pour le module basé dérivée, un deuxième seuil est présent de manière implicite lors de l'approximation des dérivées par différences finies. En terme de performance de détection, les deux méthodes donnent visuellement des résultats satisfaisants, même si le module basé dérivée est légèrement plus précis. Enfin, contrairement à la carte d'auto-occultation  $\mathcal{H}_t$ , celle introduite par le module basé dérivée peut facilement être étendue à une carte de probabilité décrite en §4.3.3.

#### 4.3.3 Carte d'auto-occultation probabiliste

L'utilisation d'une *carte probabiliste* présente de nombreux avantages : elle permet de conserver une certaine incertitude pour les pixels appartenant aux bords de la surface, et s'avère être bien mieux adaptée à un processus d'optimisation multi-résolution qu'une carte binaire comme rapporté en §4.4.3. Dans la suite du tapuscrit, nous utilisons une carte probabiliste.

Le passage à une carte probabiliste se fait en atténuant les transitions abruptes inhérentes au seuillage binaire par une fonction *sigmoïde* :

$$\psi(a; r) \stackrel{\text{def}}{=} \frac{\exp(2k(a - r))}{1 + \exp(2k(a - r))}. \quad (4.3)$$

Cette fonction dépend de deux seuils  $k$  et  $r$ . Le premier contrôle la pente de la transition (plus  $k$  est élevé plus la transition est pentue), tandis que le deuxième contrôle la position de la transition. L'influence des deux seuils est représentée sur la figure 4.6.

La carte de probabilité est alors donnée par :

$$\widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \psi \left( \min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2; r_d \right), \quad (4.4)$$

avec pour valeurs typiques  $k = 40$  et  $r_d = 0, 1$ . La figure 4.7 représente les cartes d'auto-occultation binaire et probabiliste obtenues à l'image 120 de la vidéo de la feuille de papier introduite en §4.6.

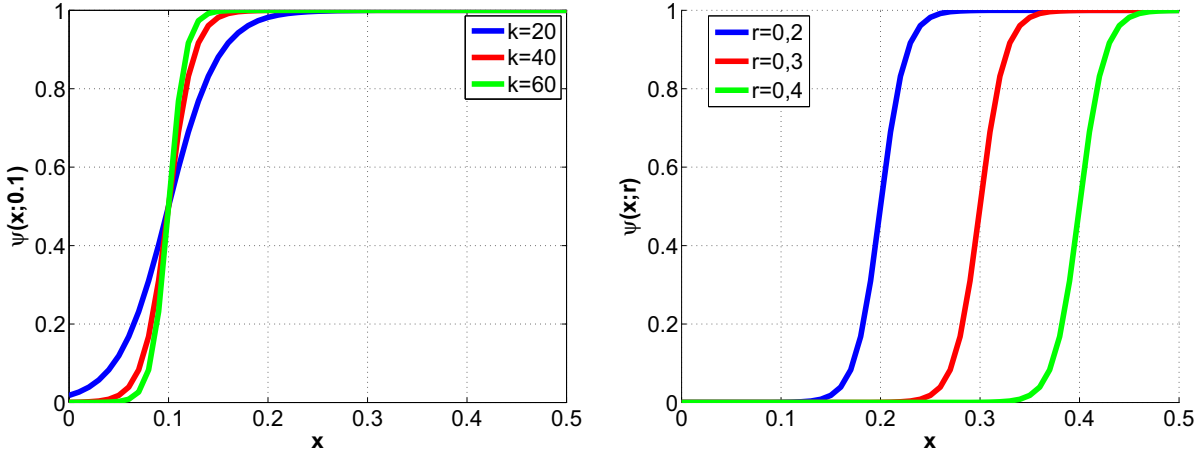


FIG. 4.6 – **Propriétés de la fonction sigmoïde utilisée.** A gauche : le paramètre  $k$  varie,  $r = 0, 1$  est fixe. A droite : le paramètre  $r$  varie,  $k = 40$  est fixe.

Notons que la fonction sigmoïde (4.3), ne peut pas être appliquée directement sur la carte  $\mathcal{H}_t$  issue du module basé position : le résultat de la détection étant fonction de plusieurs conditions.

## 4.4 Recalage non-rigide avec auto-occultations

### 4.4.1 Fonction de coût

La fonction de coût est composée de trois termes :  $\mathcal{E}_t(\mathbf{u}) = \mathcal{E}_d(\mathbf{u}) + \lambda_s \mathcal{E}_s(\mathbf{u}) + \lambda_f \mathcal{E}_f(\mathbf{u})$ . Un terme de données naturel ne prenant pas les pixels auto-occultés en compte est donné par :

$$\sum_{\mathbf{q} \in \mathcal{R}} (1 - \widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u})) \mathcal{D}^2(\mathbf{q}; \mathbf{u}). \quad (4.5)$$

Afin d'empêcher le minimum global de la fonction de coût de correspondre à une fonction de déformation entièrement comprimée, une pénalité supplémentaire  $\lambda_d(\widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}))$  doit être ajoutée. Nous privilégions néanmoins une autre approche car l'estimation simultanée de la carte d'auto-occultation et de la fonction de déformation introduit un problème d'optimisation fortement non-linéaire, sujet à de nombreux minima locaux. Nous utilisons à la place un schéma d'optimisation en deux étapes :

- ▷  $\tilde{\mathbf{u}} \leftarrow \mathbf{u}$ .  $\tilde{\mathbf{u}}$  est l'estimé du vecteur de paramètres sur l'image précédente.
- ▷ **Première étape** : mettre à jour le vecteur de paramètre en utilisant l'estimée précédente de la carte d'occultation probabiliste  $\widehat{\mathcal{H}}(\mathbf{q}; \tilde{\mathbf{u}})$ . L'énergie globale suivante est minimisée :

$$\min_{\mathbf{u}} \overbrace{\sum_{\mathbf{q} \in \mathcal{R}} (1 - \widehat{\mathcal{H}}(\mathbf{q}; \tilde{\mathbf{u}})) \mathcal{D}^2(\mathbf{q}; \mathbf{u})}^{\mathcal{E}_d(\mathbf{u})} + \lambda_s \mathbf{u}^T \mathcal{Y} \mathbf{u} + \lambda_f \sum_{\mathbf{q} \in \mathcal{R}} \sum_{\mathbf{d} \in \mathcal{F}} \sum_{p \in \{x, y\}} \gamma(\mathbf{E}_l^p(\mathbf{d}; \mathbf{q}; \mathbf{u}) \mathbf{E}_r^p(\mathbf{d}; \mathbf{q}; \mathbf{u})), \quad (4.6)$$

avec  $\mathcal{Y}$  la matrice relative à l'énergie de courbure discrétisée, définie en §2.2.1.3.

- ▷ **Deuxième étape** : mettre à jour la carte d'auto-occultation probabiliste :

$$\widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) \leftarrow \psi \left( \min_{\mathbf{d} \in \mathbb{S}^1} \|\mathbf{E}(\mathbf{d}; \mathbf{q}; \mathbf{u})\|^2; r_d \right). \quad (4.7)$$

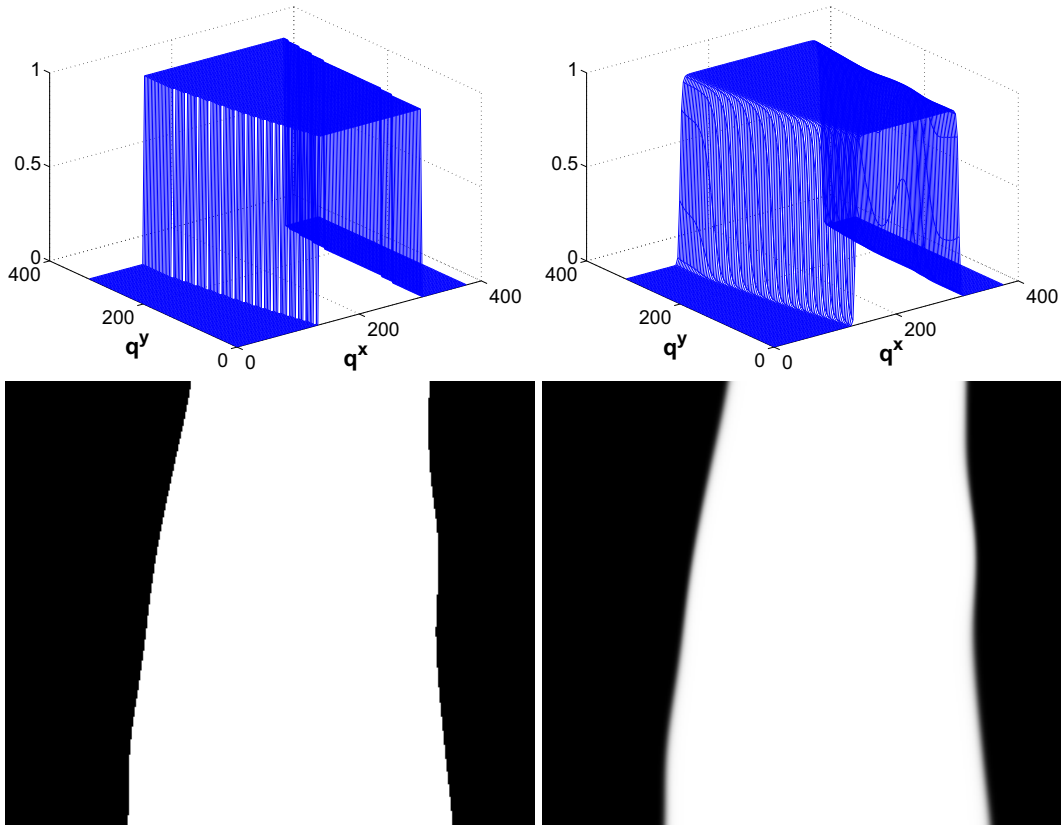


FIG. 4.7 – **Cartes d’auto-occultation binaire et probabiliste.** Illustration de la carte d’auto-occultation binaire  $\mathcal{H}(\cdot; \mathbf{u})$  (à gauche) et de la carte d’auto-occultation probabiliste  $\widehat{\mathcal{H}}(\cdot; \mathbf{u})$  (à droite). En haut : représentation 2,5D des cartes. En bas : les pixels blancs sont étiquetés comme auto-occultés et les pixels noirs étiquetés comme visibles. Les seuils sont fixés aux valeurs  $k = 40$  et  $r = r_d = 0, 1$ .

Au final, une contrainte temporelle peut également être utilisée, par exemple  $\lambda_v \|\mathbf{u} - \tilde{\mathbf{u}}\|^2$ , où  $\lambda_v$  contrôle l’influence de ce terme. Une autre approche basée sur un filtrage temporel *a posteriori*, décrit en §4.5, est adoptée.

#### 4.4.2 Occultations externes

La carte d’auto-occultation probabiliste peut être remplacée par une *carte de visibilité* probabiliste  $\widehat{\mathcal{H}}_v$ . Elle se compose de la carte d’auto-occultation probabiliste décrite en §4.3.3 et d’une *carte d’occultation externe* probabiliste  $\widehat{\mathcal{H}}_e$ . La probabilité qu’un pixel soit occulté par un objet externe est obtenue en appliquant la fonction sigmoïde  $\psi(\cdot; r_e)$  présentée en §4.3.3, à l’erreur  $\mathcal{D}(\mathbf{q}; \mathbf{u})$ . Avec cette définition, un pixel auto-occulté est la plupart du temps également détecté comme occulté par un objet externe. Nous faisons l’hypothèse qu’un pixel ne peut pas être simultanément occulté par un objet externe et auto-occulté pour construire les cartes  $\widehat{\mathcal{H}}_e$  et  $\widehat{\mathcal{H}}_v$ . Une probabilité nulle d’être occulté par un objet externe est affectée à un pixel possédant une probabilité supérieure à 0, 5 d’être auto-occulté et ceci quelque soit la valeur de  $\mathcal{D}(\mathbf{q}; \mathbf{u})$  :

$$\widehat{\mathcal{H}}_e(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \begin{cases} 0 & \widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) > 0, 5 \\ \psi(\mathcal{D}(\mathbf{q}; \mathbf{u}); r_e) & \text{sinon.} \end{cases} \quad (4.8)$$

Notons que le seuil  $r_e$  peut être fixé arbitrairement ou estimé au cours du processus d'optimisation ; il joue le même rôle que le paramètre associé aux M-estimateurs décrits en §2.2.1.4. La carte de visibilité probabiliste est obtenue en multipliant élément par élément les cartes d'auto-occultation et d'occultation externe probabilistes :

$$\widehat{\mathcal{H}}_v(\mathbf{q}; \mathbf{u}) \stackrel{\text{def}}{=} \widehat{\mathcal{H}}(\mathbf{q}; \mathbf{u}) \odot \widehat{\mathcal{H}}_e(\mathbf{q}; \mathbf{u}). \quad (4.9)$$

Des contraintes spatiales sont imposées sur la carte d'occultation externe afin de privilégier les régions denses et d'éliminer les détections ponctuelles introduites par du bruit ou des imprécisions dans le recalage. Pour cela des opérateurs morphologiques d'érosion et de dilatation sont successivement appliqués sur la carte d'occultation externe. Des expériences comprenant à la fois des occultations externes et des auto-occultations sont présentées en §4.6.

### 4.4.3 Optimisation par l'algorithme Gauss-Newton

La fonction de coût globale (4.6) est minimisée en utilisant l'algorithme DA-GN<sup>1</sup> décrit en §2.2.2.3. Les algorithmes compositionnels proposés au chapitre 3 ne peuvent pas être directement utilisés puisque l'approximation de la composition n'est plus valide pour les déformations extrêmes introduisant les auto-occultations ; comme par exemple celles représentées sur la figure 4.14.

Les FFDs<sup>2</sup> avec une interpolation par B-spline décrites en §2.3.2 sont utilisées. Elles permettent de bénéficier de matrices Jacobiennes creuses. L'emploi de méthodes prenant en compte cette propriété est nécessaire pour résoudre les équations normales ; la résolution étant sinon extrêmement coûteuse en temps de calcul. Une approche multi-résolution, présentée en §2.2.2.2, est également requise afin d'estimer précisément la déformation de la surface lorsque l'auto-occultation cesse. L'étape de raffinement inhérente à l'approche multi-résolution consiste à mettre à jour le champ de déplacement ainsi que la carte d'auto-occultation (ou de visibilité) à chaque niveau de la pyramide. Si des cartes binaires sont utilisées, cette propriété est perdue à l'issue des étapes d'expansion et de réduction car elles retournent des valeurs réelles. Trois opérations sont alors possibles afin de récupérer une carte binaire à chaque niveau de la pyramide : les labels différents de 1 sont mis à 0, les labels compris entre 0 et 1 sont fixés à la valeur entière la plus proche et enfin tous les labels différents de 0 sont mis à 1. En pratique, la dernière possibilité est à privilégier. Elle surestime légèrement les régions auto-occultées, prévenant ainsi des erreurs de recalage au niveau de la frontière d'auto-occultation. Néanmoins, l'emploi de cartes probabilistes résout ce problème naturellement. Elles sont utilisées pour la plupart des expérimentations. L'évaluation des matrices Jacobiennes associées à chaque terme est décrite ci-dessous. Le tableau 4.1 récapitule les différentes étapes de l'algorithme de recalage d'images proposé, modélisant explicitement les auto-occultations.

**Le terme de données.** C'est une norme  $\mathcal{L}_2$  pondérée de la différence d'intensité  $\mathcal{D}$ . La matrice Jacobienne associée est donnée par  $\mathcal{J}_d = \text{diag}(\text{vect}(1 - \widehat{\mathcal{H}}))\mathcal{J}_{\mathcal{I}}$ . La matrice jacobienne  $\mathcal{J}_{\mathcal{I}}$  est obtenue en rassemblant les vecteurs gradients  $\mathbf{g}_{\mathcal{I}}(\mathbf{q}; \mathbf{u})$  pour chaque pixel  $\mathbf{q} \in \mathcal{R}$ . Le vecteur gradient  $\mathbf{g}_{\mathcal{I}}$  est le produit du gradient image par la matrice Jacobienne de la fonction de déformation :  $\mathbf{g}_{\mathcal{I}}(\mathbf{q}; \mathbf{u}) = \nabla \mathcal{I}^T \frac{\partial \mathcal{W}}{\partial \mathbf{u}}$ , évalués respectivement en  $\mathcal{W}(\mathbf{q}; \mathbf{u})$  et  $\mathbf{q}$ .

<sup>1</sup>DA-GN pour "Direct Additive Gauss-Newton", voir §2.2.2.3.

<sup>2</sup>"Free Form Deformations".



Initialiser le vecteur de paramètres  $\mathbf{u}$  et la carte d'auto-occultation probabiliste  $\widehat{\mathcal{H}}$  (ou la carte de visibilité probabiliste  $\widehat{\mathcal{H}}_v$ ) avec leur estimation sur l'image précédente.

▷ **Début de la boucle liée à l'approche multi-résolution :**

- *Début de la boucle des itérations de Gauss-Newton :*
  - \* Définir le contracteur  $\mathcal{E}_f$  par l'équation (4.1) étant donné  $\mathcal{W}(\cdot; \mathbf{u})$ .
  - \* Mettre à jour le vecteur de paramètres  $\mathbf{u}$  en minimisant l'équation (4.6).
  - \* Mettre à jour la carte d'auto-occultation probabiliste  $\widehat{\mathcal{H}}$  (ou la carte de visibilité probabiliste  $\widehat{\mathcal{H}}_v$ ) par (4.4) (ou (4.9)).
- *Fin de la boucle des itérations de Gauss-Newton.*
- Propagation du champ de déplacement  $\mathbf{u}$  et de la carte d'auto-occultation probabiliste  $\widehat{\mathcal{H}}$  (ou de la carte de visibilité probabiliste  $\widehat{\mathcal{H}}_v$ ) au niveau supérieur de la pyramide.

▷ **Fin de la boucle liée à l'approche multi-résolution.**

TAB. 4.1 – Les différentes étapes de l'algorithme de recalage d'images d'une surface déformable en présence d'auto-occultations et / ou d'occultations externes.

**Le terme de régularisation.** La matrice Jacobienne pour le terme de régularisation est simplement donnée par  $\mathcal{J}_s = \sqrt{\lambda_s} \mathcal{Z}$ , avec  $\mathcal{Z}^\top \mathcal{Z} = \mathcal{Y}$ . Pour rappel, la matrice  $\mathcal{Y}$  est relative à l'énergie de courbure discrétisée définie en §2.2.1.3.

**Le contracteur.** La matrice Jacobienne associée est  $\mathcal{J}_f = \lambda_f \mathcal{J}_E$ . La matrice Jacobienne  $\mathcal{J}_E$  est obtenue en rassemblant pour chaque triplet  $(\mathbf{d}; \mathbf{q}; p) \in (\mathcal{F}; \mathcal{R}; \{x, y\})$  les vecteurs gradients :

$$\mathbf{g}_f(\mathbf{d}; \mathbf{q}; \mathbf{u}) = \mathbf{E}_r^p \nabla \mathbf{E}_l^p + \nabla \mathbf{E}_r^p \mathbf{E}_l^p,$$

pour lesquels les termes  $\mathbf{E}_l^p(\mathbf{d}; \mathbf{q}; \mathbf{u}) \mathbf{E}_r^p(\mathbf{d}; \mathbf{q}; \mathbf{u})$  associés sont négatifs.

**Mise à jour du vecteur de paramètres.** L'incrément  $\Delta \mathbf{u}$  des paramètres est obtenu en résolvant les équations normales, introduites en §1.6.1, par des méthodes prenant en compte la nature extrêmement creuse des matrices mises en jeu. La matrice Jacobienne associée à l'équation (4.6) est obtenue en rassemblant les matrices Jacobiennes de chaque terme :

$$\mathcal{J}_t^\top = \begin{pmatrix} \mathcal{J}_d^\top & \mathcal{J}_s^\top & \mathcal{J}_f^\top \end{pmatrix}.$$

Sa nature creuse s'explique en partie par l'aspect local du noyau d'interpolation utilisé par les FFD. Le résidu associé à un pixel  $\mathbf{q}$  ne dépend que des points de contrôle de son voisinage. Pour une interpolation par B-spline cubique, seuls 16 points de contrôle influencent la position de  $\mathcal{W}(\mathbf{q}; \mathbf{u})$  ce qui implique que la matrice Jacobienne  $\mathcal{J}_d$  n'a que 32 composantes non nulles par ligne. La nature creuse de la matrice Jacobienne et de l'approximation de Gauss-Newton  $\mathcal{J}_t^\top \mathcal{J}_t$  de la matrice Hessienne est illustrée sur la figure 4.8. Elles n'ont pas de structure spécifique, contrairement à la structure par blocs des matrices Jacobienne et Hessienne issues du processus de reconstruction 3D de scène rigide par ajustement de faisceaux. La prise en compte de cette nature creuse est néanmoins indispensable

pour pouvoir résoudre les équations normales en raison de la taille importante de la matrice Hessienne. Nous utilisons le module de factorisation de Cholesky directement disponible dans le logiciel Matlab<sup>3</sup>.

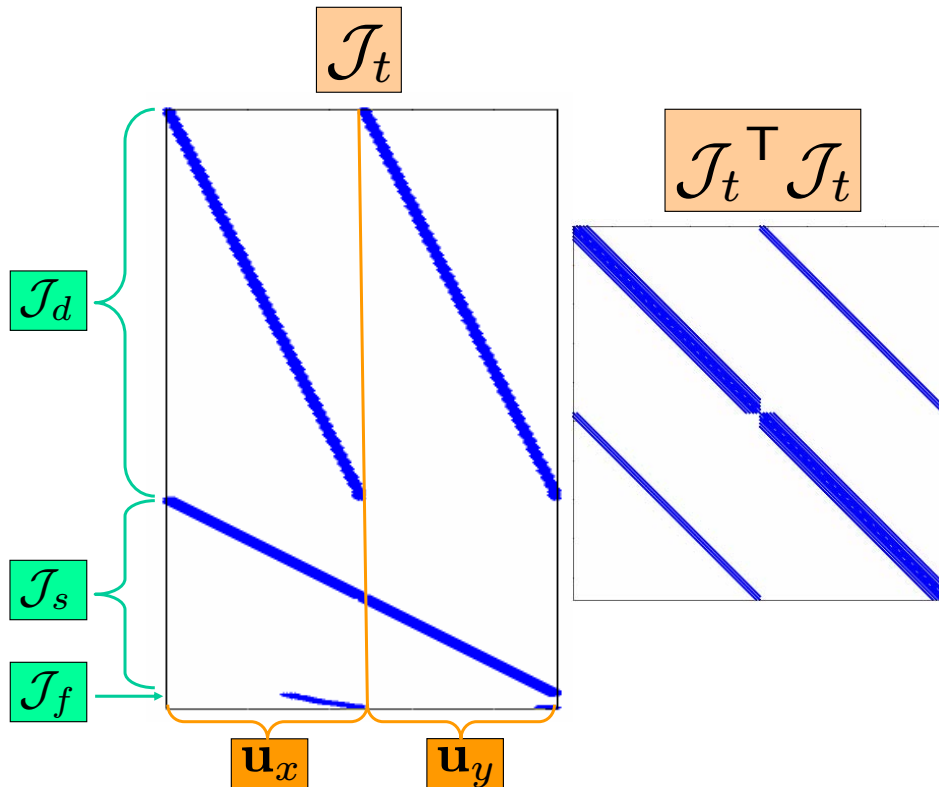


FIG. 4.8 – Structures des matrices Jacobienne et Hessienne. Les matrices Jacobienne et Hessienne (approximation de Gauss-Newton) ont une structure très creuse. Seules les lignes de  $\mathcal{J}_f$  ayant au moins un élément non-nul sont représentées. Ces matrices sont obtenues lors d'une itération du traitement de la vidéo de la feuille de papier présentée en §4.6.

## 4.5 Augmentation 2D d'images d'une surface auto-occultée

Une fois la fonction de déformation estimée pour chaque image d'une vidéo, l'augmentation 2D réaliste de cette dernière n'est pas pour autant triviale. Tout d'abord, des « tremblements » peuvent apparaître au niveau du recalage lorsque l'auto-occultation est prépondérante, puisque l'information résiduelle contenue dans les images est alors très faible. Nous comparons différentes méthodes permettant d'atténuer ces vibrations sans pour autant dégrader la précision du recalage. Elles consistent à régulariser ou à filtrer les trajectoires des points de contrôle. Pour la première méthode, des splines cubiques sont utilisées. Pour la deuxième, des filtres Gaussien et médian sont étudiés. Ensuite, le choix de la méthode d'augmentation est primordial pour un rendu réaliste, notamment au niveau des pixels proches de la frontière d'auto-occultation. Celui-ci est directement lié à la représentation des auto-occultations.

<sup>3</sup>Plus précisément, nous utilisons la fonction "mldivide" de Matlab.

**Contraintes temporelles.** Une possibilité est de filtrer les trajectoires  $\mathbf{t}_i$  des points de contrôle :  $\mathbf{t}_i^f = \mathbf{t}_i * \vartheta$ . Un filtre linéaire Gaussien  $\frac{1}{\sqrt{2\pi\sigma}} \exp(-\frac{a}{2\sigma^2})$  et le filtre non-linéaire médian, sont utilisés. Ce dernier estime la valeur médiane sur une fenêtre d'intérêt. La taille des filtres utilisés au cours des expérimentations est de 7 sur l'axe temporel. La deuxième approche régularise les trajectoires par des splines cubiques. La courbe régularisée  $\mathbf{t}_i^r$  qui passe au mieux par  $\mathbf{t}_i$  est estimée. Ceci revient à minimiser la fonction de coût suivante :  $\|\mathbf{t}_i - \mathbf{t}_i^r\|^2 + \|\mathcal{Z}\mathbf{t}_i^r\|^2$ , où  $\mathcal{Z}$  est une matrice de régularisation assurant la minimisation de l'énergie de courbure discrétisée, définie en §2.2.1.3.

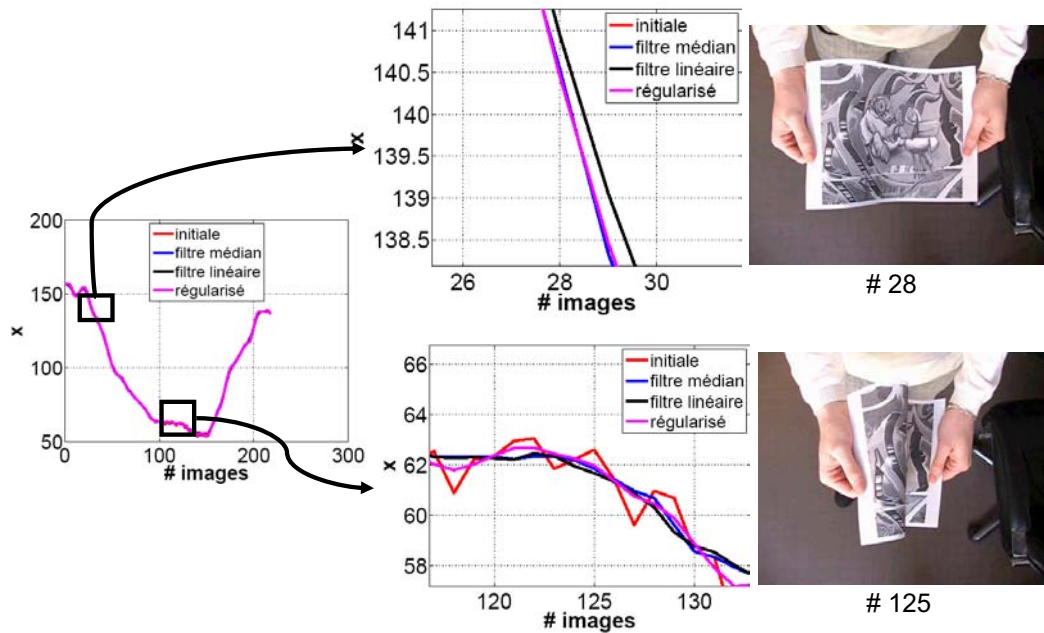


FIG. 4.9 – **Contraintes temporelles sur la trajectoire des points de contrôle.** La taille des filtres utilisée est de 7 sur l'axe temporel. A gauche : la trajectoire globale. En haut à droite : zoom sur une portion sans tremblement et une image associée. La trajectoire initiale et celle obtenue après filtrage médian ne sont pas discernables sur cette portion. En bas à droite : zoom sur une portion avec tremblements et une image associée. Les effets des filtrages et de la régularisation sont clairement visibles.

La figure 4.9 illustre les résultats obtenus par les différentes approches appliquées sur la trajectoire d'un point de contrôle (composante  $x$ ) de la vidéo de la feuille de papier. Des tremblements ont lieu entre les images 100 et 150 lorsque l'auto-occultation est prépondérante. La régularisation ainsi que les filtrages temporels atténuent ces vibrations. Cependant, les filtres linéaires modifient globalement la trajectoire. La grille de points de contrôle s'écarte de sa position idéale. Le filtrage médian et la régularisation atténuent les hautes fréquences sans pour autant dégrader la précision du recalage. Le rendu de l'augmentation 2D s'en trouve grandement amélioré. Par la suite, nous utilisons un filtrage médian. Il peut facilement être réalisé en ligne en considérant un retard de quelques images.

**Augmentation 2D d'une surface auto-occultée.** Augmenter un pixel  $\mathbf{q}_i$  peut se faire en choisissant la couleur du sous-pixel  $\mathbf{q}$  de l'image de référence le plus proche après déformation :  $\mathcal{I}(\mathbf{q}_i) \leftarrow \mathcal{I}_0(\arg \min_{\mathbf{q}} \|\mathcal{W}(\mathbf{q}; \mathbf{u}) - \mathbf{q}_i\|)$ . Cette approche n'est pas adaptée à notre représentation des auto-occultations. En effet, un grand nombre de pixels de l'image de référence sont transférés

au voisinage des pixels proches de la frontière d'auto-occultation sur l'image courante, entraînant l'apparition d'*artefacts* lors de l'augmentation.

Nous utilisons une approximation *homographique par morceaux* de la fonction de déformation inverse restreinte aux régions non auto-occultées. Elle est estimée entre les points de contrôle de l'image de référence et ceux estimés sur l'image courante. Cette transformation inverse est bien adaptée à notre représentation des auto-occultations. En effet, aucun pixel de l'image courante n'est associé aux mailles entièrement comprimées. La texture correspondante sur l'image de référence est ainsi évitée. Une illustration du processus d'augmentation 2D d'images d'une surface auto-occultée est représentée sur la figure 4.10.

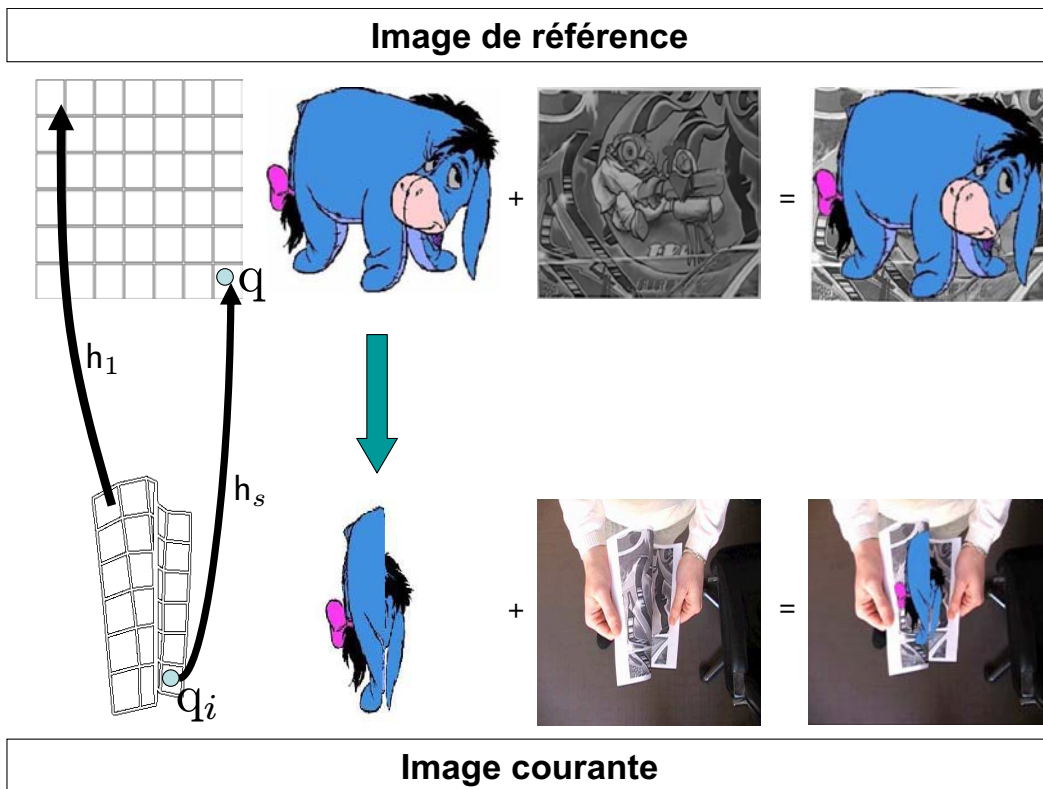


FIG. 4.10 – Illustration du processus d'augmentation 2D d'images d'une surface auto-occultée par une fonction de déformation inverse de type homographie par morceaux. Le maillage de référence et celui estimé lors du processus de recalage. Pour chacune des mailles, une homographie  $h_v$ ,  $v \in [1, \dots, s]$ , est estimée. Elle transfère un pixel de l'image courante vers l'image de référence, permettant de gérer simplement les pixels proches de la frontière d'auto-occultation. Le maillage représenté est peu dense par souci de visualisation ; celui utilisé pour le recalage est beaucoup plus dense. La fonction de déformation de type homographie par morceaux est appliqué au logo « Bourriquet » avant d'augmenter l'image courante par le procédé décrit en §1.5.2.

## 4.6 Résultats expérimentaux

Nous avons testé notre approche sur plusieurs vidéos avec différents types de surfaces (tissu, papier et tapis). Une pyramide à deux niveaux est utilisée au cours du processus multi-résolution d'op-

timisation. Elle permet d'estimer la fonction de déformation lorsque l'auto-occultation cesse (l'ajout de niveaux supplémentaires n'améliore guère le recalage). Les temps de calcul pour une image de référence composée de  $316 \times 378$  pixels<sup>4</sup> et d'une grille de  $64 \times 76$  points de contrôle sont de l'ordre de 25 secondes par image avec un code Matlab implanté sur un PC portable dont le processeur est un Intel Centrino Duo 1,84GHz. La majorité des temps de calcul est passée à former les matrices Jacobienne et Hessienne. Les équations normales sont quant à elles résolues très efficacement : environ 0,2s, grâce à la prise en compte de la nature creuse des matrices mises en jeu.

Une grille régulière est définie afin de visualiser le recalage estimé. Cette dernière est moins dense que la grille de points de contrôle. Pour illustrer la détection des pixels auto-occultés nous avons surimposé à l'image de référence la carte d'auto-occultation probabiliste estimée pour chaque image :

$$\mathcal{I}_0 \leftarrow \text{saturate}(\mathcal{I}_0 + 255\widehat{\mathcal{H}}; 255),$$

avec

$$\text{saturate}(a; 255) \stackrel{\text{def}}{=} \begin{cases} a & \text{si } a \leq 255 \\ 255 & \text{sinon.} \end{cases}$$

Un pixel ayant une probabilité égale à 1 (c'est-à-dire auto-occulté) est représenté en blanc et un pixel ayant une probabilité égale à 0 conserve son intensité d'origine.

Il y a un certain nombre de seuils présents dans la fonction de coût : les poids associés aux termes de régularisation  $\lambda_s$  et au contracteur  $\lambda_f$  et les paramètres associés au module de détection,  $r_d$  et  $k$ . Seul le paramètre  $\lambda_s$  varie au cours des différentes expériences, les autres restent constants. Des solutions sont proposées pour sélectionner automatiquement ce paramètre de régularisation (Bartoli, 2007) avec un critère basé primitive. L'extension de cette approche à un critère direct n'est pas triviale et devra faire l'objet d'études supplémentaires.

Notons que l'algorithme de recalage proposé gère également la disparition des pixels sous l'effet de la projection perspective. Dans ce cas, les modules de détection proposés en §4.3.2.1, §4.3.2.2 et §4.3.3, sont toujours valides et donnent de bons résultats. Le contracteur n'est cependant pas activé puisque la fonction de déformation ne tend pas à se rabattre sur elle même mais se comprime naturellement.

### 4.6.1 Le contracteur

En l'absence de ce terme, la fonction de déformation a naturellement tendance à former une boucle, comme l'illustre la figure 4.11. Les définitions d'un pixel auto-occulté introduites en §4.3.2 ne sont alors plus valides, la carte d'auto-occultation est mal estimée, perturbant le recalage. L'ajout de ce nouveau terme dans la fonction de coût contraint la fonction de déformation à se comprimer le long de la frontière d'auto-occultation. Les modules de détection proposés sont alors valides, la carte d'auto-occultation devient dense et précise.

<sup>4</sup>Tous les pixels sont pris en compte.

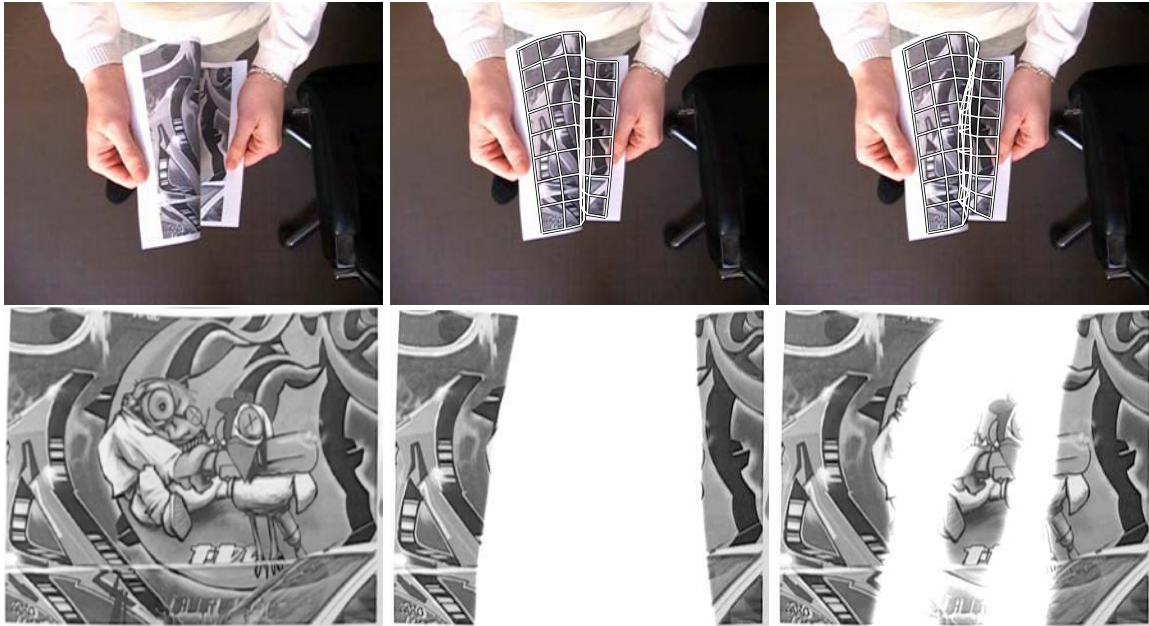


FIG. 4.11 – **Exemples de recalage d’images avec ou sans contracteur.** A gauche : l’image courante  $\mathcal{I}$  et celle de référence  $\mathcal{I}_0$ . Au milieu : résultat du recalage et détection des régions auto-occultées avec le contracteur. A droite : résultat du recalage et détection des régions auto-occultées sans le contracteur.

## 4.6.2 Une vidéo synthétique

### 4.6.2.1 Description

Cette vidéo constituée de 300 images est réalisée en utilisant le modèle 3D de feuille de papier décrit dans (Perriollat and Bartoli, 2007)<sup>5</sup>. Pour chaque image de la vidéo, une carte d’auto-occultation binaire  $\mathcal{H}_f$  ainsi qu’un ensemble de points en correspondance sont disponibles, comme l’illustre la figure 4.12. Ces derniers sont obtenus par projection des maillages 3D ayant généré les déformations. Les cartes d’auto-occultation sont quant à elles estimées par la technique du z-buffer. La vérité terrain sur cette vidéo est donc connue. L’auto-occultation apparaît à l’image 53 et disparaît à l’image 198. Au moins 45% de la surface est auto-occultée de l’image 100 à 135. Ces caractéristiques sont précisément déterminées par la présence de valeurs non-nulles dans  $\mathcal{H}_f$ .

### 4.6.2.2 Résultats

La précision de la détection des auto-occultations est évaluée en mesurant le pourcentage de pixels qui ne sont pas correctement détectés. Le module de détection binaire basé sur la dérivée, décrit en §4.3.2.2, est utilisée puisque les cartes d’auto-occultation de référence  $\mathcal{H}_f$  sont binaires. La qualité du recalage, exprimé en pixels, est mesurée par le RMS<sup>6</sup> entre la projection du maillage 3D ayant induit l’image synthétique et sa position estimée par notre algorithme. Seuls les sommets correspondant à des régions visibles de la surface<sup>7</sup> sont pris en compte dans l’estimation de l’erreur. Les résultats

<sup>5</sup>Cette vidéo a été réalisée par Mathieu Perriollat.

<sup>6</sup>"Root Mean Square".

<sup>7</sup>Les cartes  $\mathcal{H}_f$  sont utilisées pour définir les régions visibles de la surface

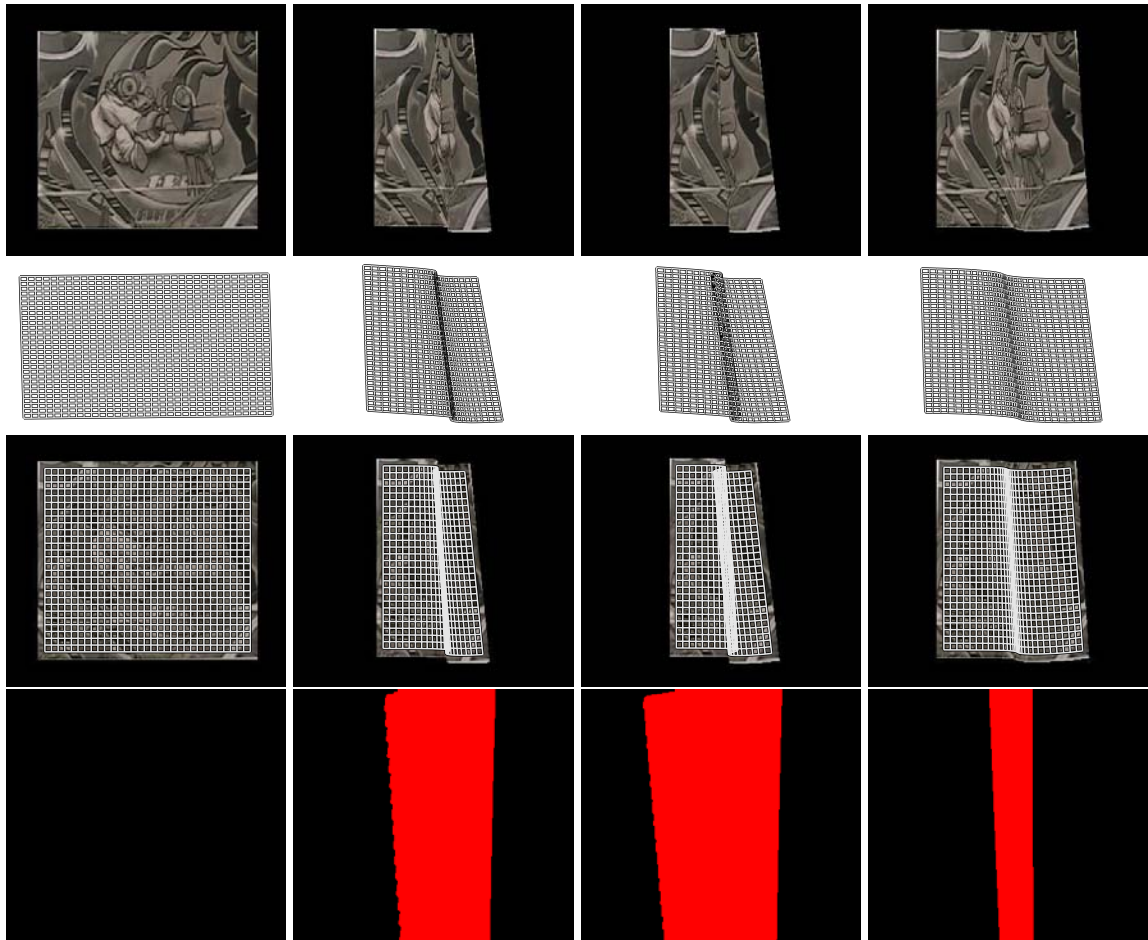


FIG. 4.12 – **Illustration de la vidéo synthétique et de la vérité terrain associée.** PEn haut : images extraites de la vidéo synthétique. 2<sup>ème</sup> ligne : les formes 3D ayant généré les images. 3<sup>ème</sup> ligne : les correspondances de points utilisées comme vérité terrain. En bas : les cartes d'auto-occultation utilisées comme vérité terrain : un pixel rouge (gris) est auto-occulté.

obtenus sont représentés sur la figure 4.13.

L'erreur résiduelle moyenne au cours de la vidéo est de 0,3 pixels, traduisant une estimation précise des parties visibles et de la déformation lorsque l'auto-occultation cesse. Un pic apparaît lorsque l'auto-occultation est à son apogée, en raison de légères imprécisions de recalage au niveau de la frontière d'auto-occultation lorsque celle-ci devient importante. Le maximum n'est néanmoins que de 1,2 pixels.

La détection des auto-occultations est suffisamment précise pour assurer la qualité du recalage. En moyenne, 98% des pixels sont correctement labélisés. Dans le pire des cas ils sont tout de même 92%. Des pics apparaissent lorsque l'auto-occultation est extrême mais également au début et à la fin de l'auto-occultation. Notre module de détection anticipe les auto-occultations puisque nous avons délibérément fixé le seuil  $r_d$  de l'équation (4.2) à une valeur légèrement supérieure à 0. Par exemple, sur l'image 52 de la vidéo, la surface n'est pas encore auto-occultée que notre module de détection s'active pour quelques pixels. Ces derniers ne sont réellement auto-occultés qu'à partir de l'image suivante.

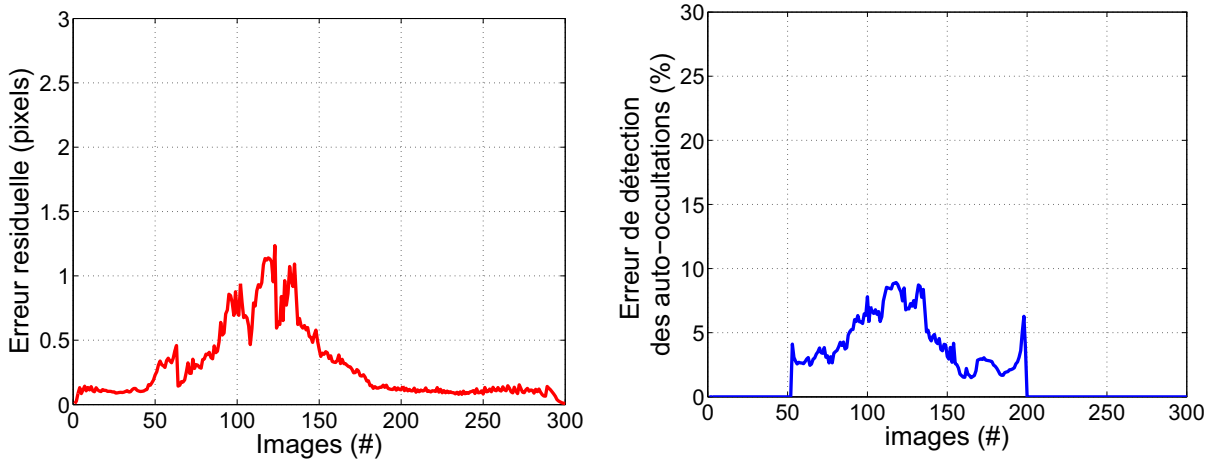


FIG. 4.13 – **Résultat du recalage et de la détection des auto-occultations sur la vidéo synthétique.** A gauche : l'erreur résiduelle (RMS en pixels). A droite : pourcentage d'erreur sur la détection des auto-occultations.

### 4.6.3 La vidéo de la feuille de papier

Nous donnons le détail du traitement de cette vidéo et des résultats obtenus.

#### 4.6.3.1 Description de la vidéo

Elle est composée de 215 images de taille  $720 \times 576$ . L'auto-occultation apparaît visuellement aux alentours de l'image 50 et disparaît vers l'image 180. Elle est extrêmement importante puisqu'entre les images 80 et 160 environ 50% de la surface est auto-occultée. L'image de référence  $\mathcal{I}_0$  est de taille  $316 \times 378$ . Le nombre de points de contrôle est 4.864. Ils sont organisés sur une grille de dimension  $64 \times 76$ . Quelques images de la vidéo de la feuille de papier sont représentées sur la figure 4.14. Un grand nombre de points de contrôle est choisi lors des expérimentations pour obtenir une transformation flexible. Cette propriété est notamment requise dans les régions auto-occultées. En contrepartie, la fonction de déformation est fortement régularisée afin d'être suffisamment contrainte.



FIG. 4.14 – Images extraites de la vidéo de la feuille de papier.

#### 4.6.3.2 La matrice Jacobienne

La taille de la matrice Jacobienne  $\mathcal{J}_d$  est  $119.248 \times 9.728$  ; le nombre d'éléments non-nuls est 3.815.936 soit environ 0,33%. La matrice Jacobienne  $\mathcal{J}_s$  associée au terme de régularisation est de



taille  $9.728 \times 9.728$ , le nombre de composantes non-nulles est de 123.672, soit environ 0,13%. Enfin, la matrice Jacobienne associée au contracteur  $\mathcal{J}_f$  est nulle, sauf en présence d'auto-occultations, comme l'illustre la figure 4.15. La région en pointillé correspond aux images pour lesquelles le contracteur est activé pour estimer la déformation. Sur cet exemple, il est activé entre les images 61 et 170.

### 4.6.3.3 Détection des auto-occultations

Nous utilisons le module basé sur la dérivée et plus précisément la carte de probabilité associée. Ce choix est discuté en §4.3.3. La figure 4.15 représente l'évolution du pourcentage de pixels détectés comme auto-occultés. Nous définissons un pixel auto-occulté si sa probabilité associée est supérieure à 0,5. La présence de régions auto-occultées est décelée entre les images 46 et 184, ce qui correspond approximativement à nos observations. Le module anticipe légèrement les auto-occultations, comme pressenti. Le pourcentage maximum de pixels détectés comme auto-occultés est aux alentours de 50% lorsque l'auto-occultation est à son apogée ; ce qui est également en accord avec nos observations. D'un point de vue général, la détection des régions auto-occultées apparaît visuellement cohérente et précise. La figure 4.15 illustre également des exemples de détection sur cette vidéo.

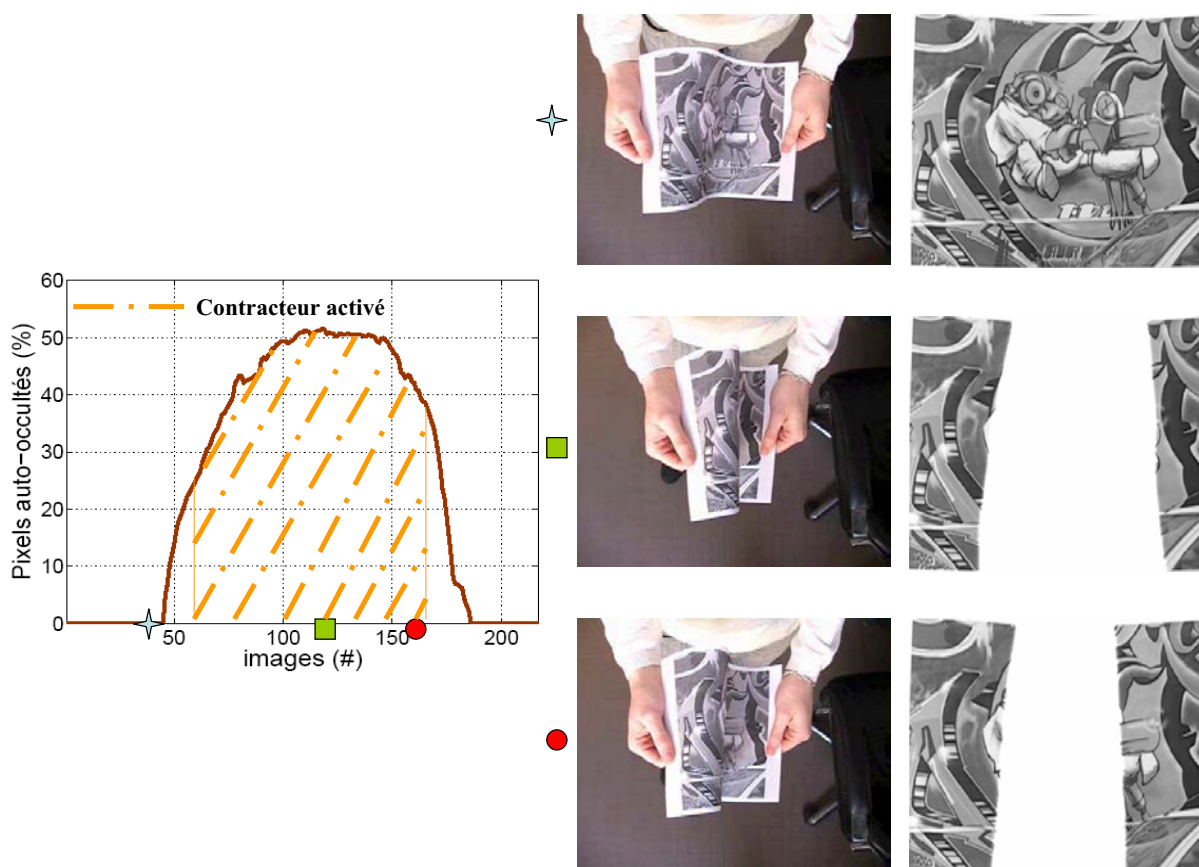


FIG. 4.15 – Détection des pixels auto-occultés sur la vidéo de la feuille de papier. A gauche : le pourcentage de pixels détectés comme auto-occultés. A droite : certaines images de cette vidéo ainsi que les résultats de la détection.

#### 4.6.3.4 Résultat du recalage et de l'augmentation 2D

Le recalage des régions visibles est précis tandis que la fonction de déformation se comprime bien au niveau de la frontière d'auto-occultation. Son estimation lorsque l'auto-occultation cesse se fait sans erreur de recalage. Des tremblements apparaissent lorsque l'auto-occultation devient extrême comme nous l'avons préalablement rapporté. Ils sont sensiblement atténués par le module de filtrage temporel présenté en §4.5 mais ne disparaissent pas totalement. Le rendu de l'augmentation 2D est très satisfaisant sur cette vidéo, comme l'illustre la figure 4.16. La tête, certaines pattes ainsi qu'une partie du corps du logo « Bourriquet » disparaissent et réapparaissent au rythme des déformations de la surface. Quelques légères imprécisions surviennent néanmoins le long de la frontière d'auto-occultation lorsque celle-ci est prépondérante.

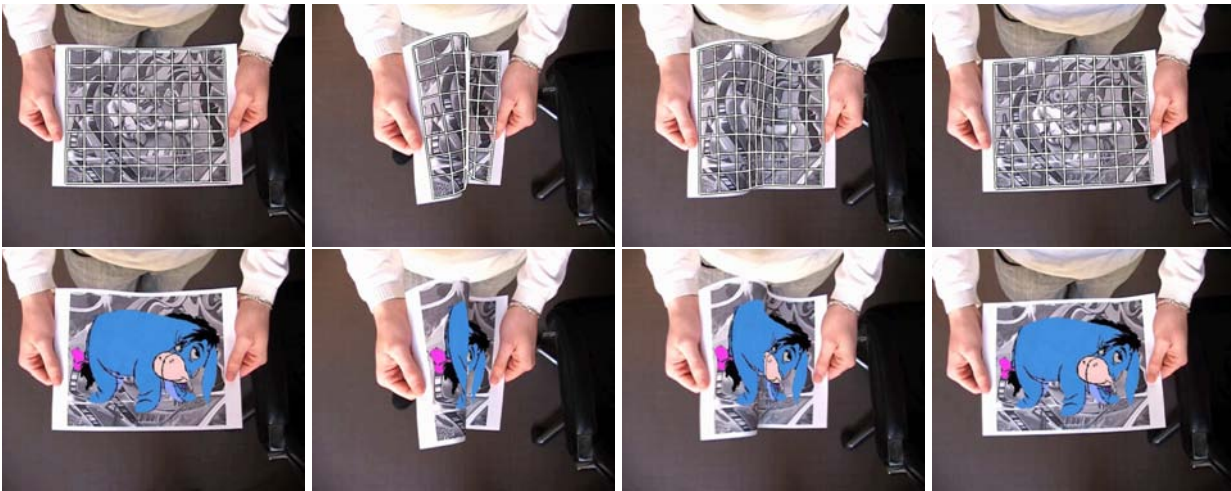


FIG. 4.16 – Résultat du recalage et de l'augmentation 2D sur la vidéo de la feuille de papier. En haut : déformations estimées. En bas : les images augmentées avec un logo « Bourriquet ».

#### 4.6.4 Comparaison avec les méthodes robustes

Nous avons comparé, sur la deuxième vidéo de la feuille de papier, notre approche avec les méthodes classiques de recalage directes robustes, décrites en §2.2.1.4. Cette vidéo se compose de 253 images de taille  $720 \times 560$ . L'auto-occultation apparaît entre les images 89 et 191. Elle est prépondérante : plus de la moitié de la surface disparaît.

Nous rappelons que les méthodes directes robustes minimisent l'équation (2.2) par l'algorithme DA-GN, décrit en §2.2.2.3, en y incluant un estimateur robuste. Plusieurs noyaux robustes ont été testés et donnent sensiblement les mêmes résultats. Ceux présentés sur la figure 4.17 sont obtenus avec la fonction de Huber décrite en §2.2.1.4. Afin d'être consistant, un schéma d'optimisation multi-résolution est également appliqué. Notre approche estime avec succès les déformations de la surface tandis que les méthodes classiques n'y parviennent pas. En effet, la détection des pixels auto-occultés étant imprécise, des erreurs de recalage apparaissent, notamment aux niveaux des régions encore apparentes. Il devient alors peu probable d'estimer sans erreur la déformation lorsque l'auto-occultation cesse.

Notons qu'une procédure permettant de réinitialiser le recalage peut être utilisée lorsque celui-ci est perdu, comme par exemple (Pilet et al., 2008). Notre approche ne requiert pas de telles procédures.

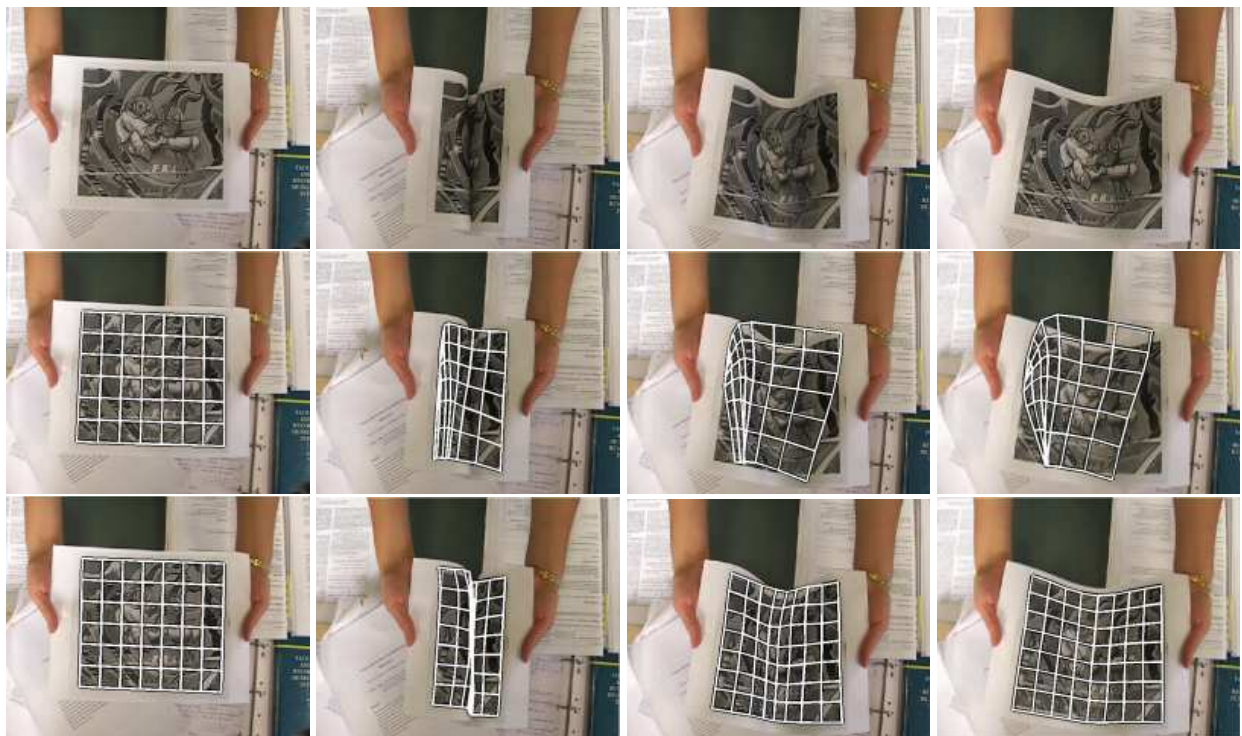


FIG. 4.17 – **Résultat du recalage sur la deuxième vidéo de la feuille de papier.** En haut : images extraites de la vidéo. Au milieu : résultat du recalage avec les méthodes robustes classiques ne gérant pas explicitement les auto-occultations. En bas : résultat du recalage avec la méthode proposée.

### 4.6.5 Auto-occultation sur les bords de la surface

Nous avons également testé le comportement de notre approche lorsque l'auto-occultation apparaît sur les bords de la surface. Deux vidéos sont réalisées, respectivement appelées "boundary I" et "boundary II". Elles sont respectivement constituées de 204 et 277 images. Pour ces deux vidéos, l'auto-occultation est assez importante : environ un quart de la surface est caché.

Ce type d'auto-occultation constitue un cas particulier dans le sens où les méthodes classiques et celle proposée sont toutes valides : les pixels non-visibles sont correctement rejetés et ceux visibles correctement recalés. Cependant, la frontière d'auto-occultation est suivie avec notre approche comme l'illustre la figure 4.18. La minimisation de l'équation (4.6) force la fonction de déformation à se comprimer le long de la frontière d'auto-occultation. La frontière d'auto-occultation est perdue par les méthodes classiques. L'erreur résiduelle (RMS) en terme d'intensité, pour ces deux vidéos, est représentée sur la figure 4.19. Les valeurs obtenues sont acceptables en dépit de déformations et d'auto-occultations importantes. Pour les vidéos "boundary I" et "boundary II", le RMS moyen se situe respectivement aux alentours de 11 UI<sup>8</sup> et 16 UI. Des erreurs résiduelles faibles pour les dernières images : moins de 12 UI et moins de 10 UI respectivement pour les vidéos "boundary I" et "boundary II" prouvent que les déformations sont correctement estimées lorsque l'auto-occultation cesse.

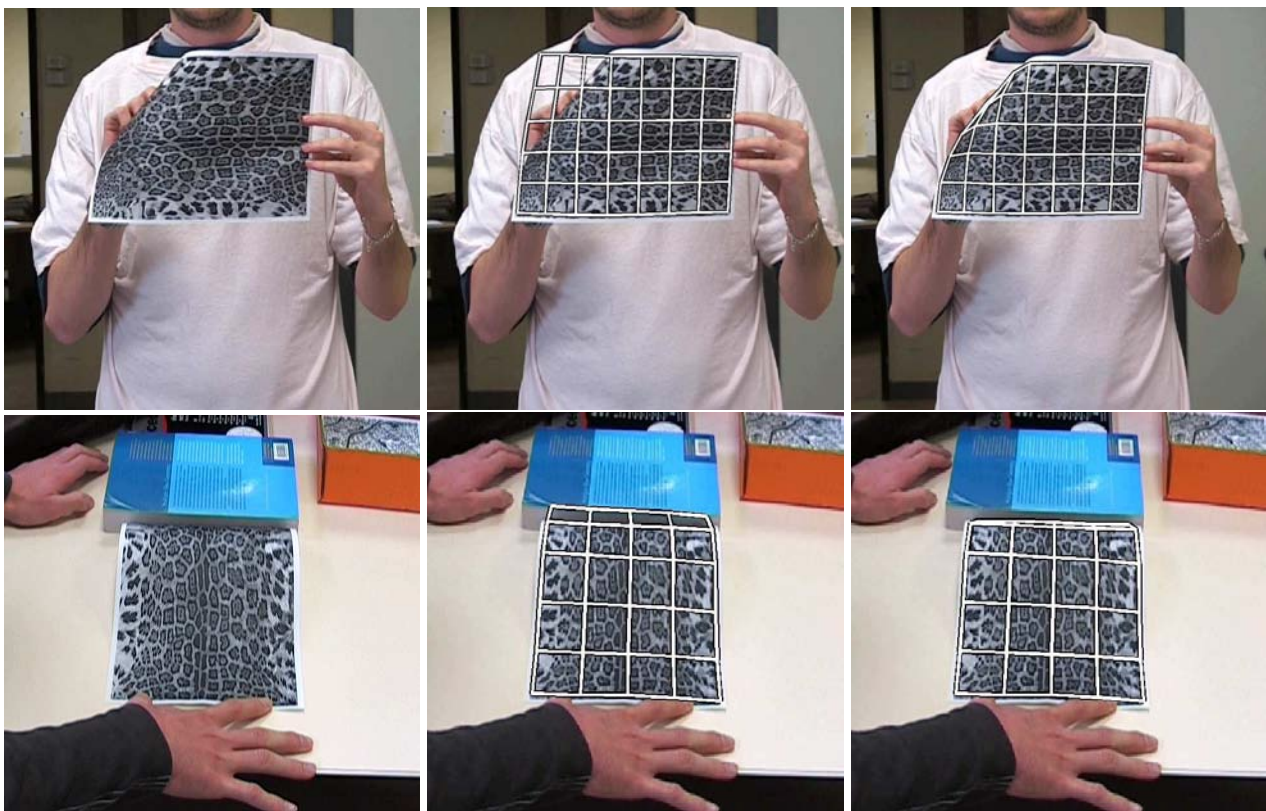


FIG. 4.18 – Résultat du recalage sur des vidéos présentant des surfaces auto-occultées sur les bords. A gauche : en haut, une image de la vidéo "boundary I" et en bas, une image de la vidéo "boundary II". Au milieu : résultat du recalage avec des méthodes robustes classiques ne gérant pas explicitement les auto-occultations. A droite : résultat du recalage avec la méthode proposée.

<sup>8</sup>Unité d'Intenisté.

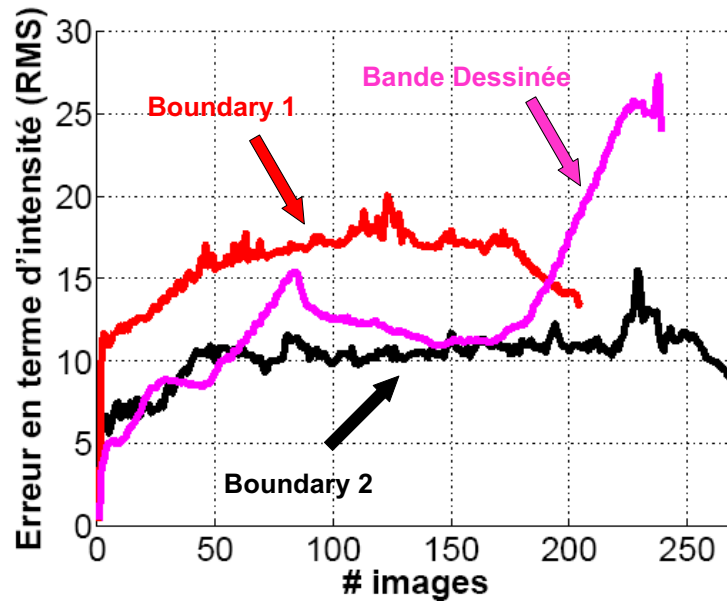


FIG. 4.19 – Erreurs résiduelles (RMS) en terme d'intensité pour différentes vidéos.

#### 4.6.6 Occultations externes

L'approche proposée est également robuste aux occultations externes de par la carte de visibilité et le terme de régularisation. La figure 4.20 représente des images de la troisième vidéo de la feuille de papier sur laquelle les deux types d'occultations se produisent. Cette vidéo se compose de 217 images. L'auto-occultation ainsi que l'occultation externe apparaissent et disparaissent simultanément aux images 37 et 160.

Chaque pixel est étiqueté comme visible, occulté par un objet externe ou auto-occulté. Pour les pixels visibles, la fonction de déformation est lisse et  $\widehat{\mathcal{H}}_v(\cdot; \mathbf{u}) < 0,5$ , pour ceux occultés par un objet externe, la fonction de déformation est également lisse et  $\widehat{\mathcal{H}}_e(\cdot; \mathbf{u}) > 0,5$ . Enfin, pour les pixels auto-occultés, la fonction de déformation est comprimée et  $\widehat{\mathcal{H}}(\cdot; \mathbf{u}) > 0,5$ .

#### 4.6.7 La vidéo de la bande dessinée

Cette vidéo se compose de 255 images. La page suivie disparaît totalement à la fin de la vidéo, comme l'illustre la figure 4.21. L'auto-occultation est totale. L'algorithme proposé augmente de manière convaincante les régions de la surface encore apparentes. A la fin de la vidéo, la fonction de déformation est entièrement comprimée et la carte d'auto-occultation est pleine, c'est-à-dire que tous les pixels ont une probabilité supérieure à 0,5 d'être auto-occultés. Le logo "Cars", utilisé pour l'augmentation 2D, disparaît naturellement à la fin de la vidéo. Notons qu'un algorithme de détection de surfaces déformables, par exemple (Pilet et al., 2008), permet d'augmenter une nouvelle page lorsque la précédente disparaît entièrement. La combinaison de cette approche avec celle proposée est à envisager dans des travaux futurs. L'erreur résiduelle (RMS) en terme d'intensité associée à cette vidéo est représentée sur la figure 4.19. Elle est légèrement supérieure à 10 UI pour les images comprises entre 100 et 200 pour lesquelles la surface est auto-occultée à 50%. Sur la fin de la vidéo, c'est-à-dire aux alentours de l'image 210, le RMS passe à 20 UI, les régions encore visibles sont un peu moins

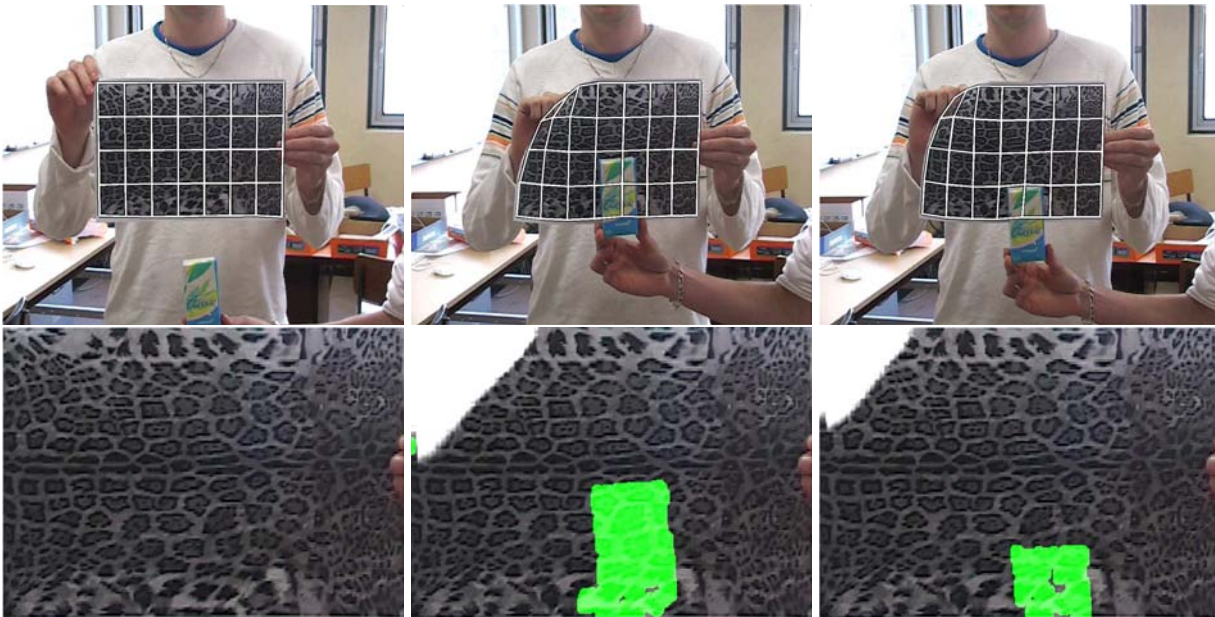


FIG. 4.20 – **Résultat du recalage sur une vidéo présentant une occultation externe et une auto-occultation.** En haut : déformations estimées. En bas : l'image de référence  $\mathcal{I}_0$  avec les pixels détectés comme auto-occultés en blanc et ceux détectés comme occultés par un objet externe en vert (gris clair).

bien recalées, la surface étant auto-occultée à plus de 60%. De plus, des variations d'éclairément apparaissent en raison des fortes déformations, expliquant en partie les valeurs importantes de l'erreur résiduelle. Pour les dernières images, elle n'existe plus puisque tous les pixels sont auto-occultés.

#### 4.6.8 Autres types de surfaces

Nous avons également testé notre approche sur un tapis et sur un tissu. Pour ce dernier, la vidéo associée est augmentée. Le rendu obtenu est très réaliste comme l'illustre la figure 4.23. En effet, une partie de la tête du logo "Mickey" ainsi que l'une de ses jambes disparaissent pendant l'auto-occultation et réapparaissent très naturellement lorsque celle-ci cesse. Le résultat du recalage sur la vidéo du tapis est représenté sur la figure 4.22.

#### 4.6.9 Echecs

L'approche proposée échoue lorsque les déformations de la surface vont à l'encontre du terme de régularisation (l'énergie de courbure). Par exemple, une vidéo d'une feuille de papier que l'on froisse ou encore un T-shirt présentant de multiples pliures font échouer notre algorithme. Les figures 4.24(a) et 4.24(b) illustrent ces configurations.

### 4.7 Conclusion

Nous avons présenté une approche originale pour le recalage d'images d'une surface déformable auto-occultée. Le cadre proposé s'appuie sur deux composantes principales. Tout d'abord, la fonction

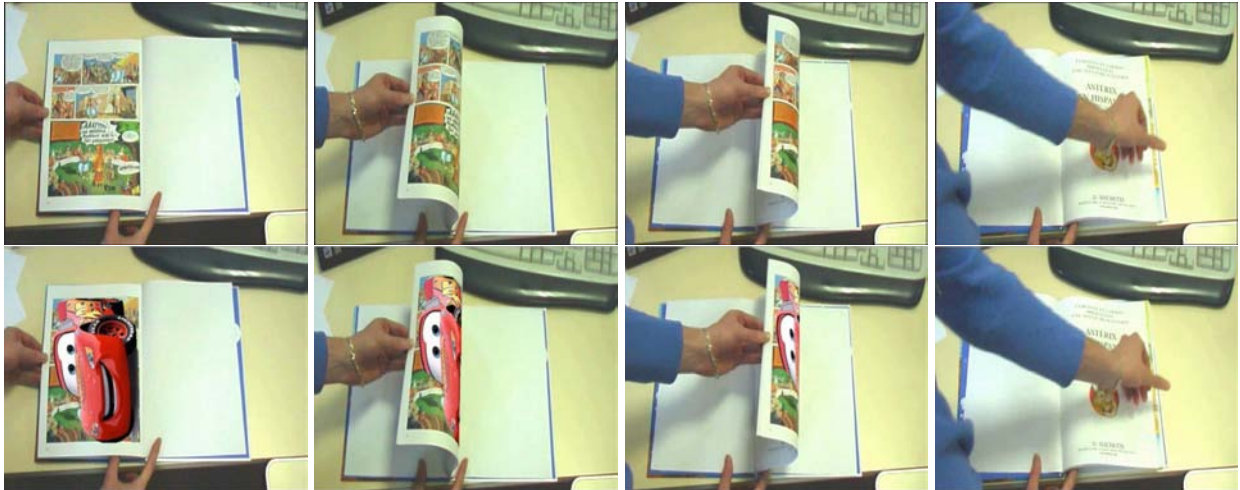


FIG. 4.21 – **Augmentation 2D de la vidéo bande dessinée.** En haut : les images originales. En bas : les images augmentées avec le logo "Cars".

de déformation est contrainte à se comprimer le long de la frontière d'auto-occultation par l'ajout d'un nouveau terme à la fonction de coût classique, appelé le contracteur. Puis cette propriété de contraction est utilisée pour définir un module de détection de pixels auto-occultés. Il se base sur l'étude des dérivées directionnelles évaluées en chaque pixel de l'image de référence et conduit à une carte d'auto-occultation probabiliste.

Les résultats expérimentaux sur des vidéos réelles prouvent que notre approche est clairement mieux adaptée que les méthodes robustes classiques. En effet, le recalage des régions apparentes est très précis, ce qui permet d'estimer correctement les déformations lorsque l'auto-occultation cesse. La détection des régions auto-occultées est en outre dense et proche de la réalité. Sur ces vidéos, les méthodes classiques échouent dès que l'auto-occultation devient significative. Nous avons étendu notre méthode à l'augmentation 2D d'images d'une surface déformable pour des applications de post-production. Le rendu obtenu, notamment au niveau de la frontière d'auto-occultation, est très gratifiant.

Afin d'aboutir à un système complet et parfaitement réaliste, quelques améliorations doivent être apportées. Tout d'abord les variations d'illumination ne sont pas prises en compte. Elles engendrent des gênes visuelles dégradant le rendu réaliste de l'augmentation. De plus, la méthode de détection des occultations externes utilisée n'est pas suffisamment précise pour ces applications. Nous envisageons d'adapter les travaux existants (Bradley and Roth, 2004; Pilet et al., 2007) sur la détection des occultations externes et des variations d'illumination, à notre méthode de recalage afin d'aboutir à un système plus complet. L'amélioration des temps de traitement ainsi que la sélection automatique des différents paramètres sont également des contraintes qui devront être prises en compte.

Le champ de déplacement dense obtenu en sortie de l'algorithme de recalage d'images que nous venons de présenter, est utilisé, après sous échantillonnage, comme donnée d'entrée de l'algorithme de reconstruction 3D de surfaces déformables présenté au chapitre suivant.



FIG. 4.22 – Résultat du recalage sur la vidéo du tapis. En haut : images extraites de la vidéo. En bas : déformations estimées.



FIG. 4.23 – Augmentation 2D de la vidéo du tissu. En haut : les images originales. En bas : les images augmentées avec un logo "Mickey".



(a)

(b)

FIG. 4.24 – Déformations faisant échouer l'approche proposée. (a) Une feuille de papier froissée. (b) Un Tshirt présentant de multiples pliures.



# Chapitre 5

## Reconstruction 3D de surfaces déformables

---

---

Dans ce chapitre, nous abordons le problème de la reconstruction 3D de surfaces déformables à partir de points mis en correspondance sur plusieurs images. Nous utilisons le modèle de faible rang : les déformations 3D sont modélisées par une combinaison linéaire de modes de déformation. Une nouvelle manière de représenter ce modèle est proposée : les modes de déformation sont ordonnés en fonction de l'amplitude des déformations qu'ils capturent. Cette représentation présente de nombreux avantages. Elle permet de lever certaines ambiguïtés et introduit un algorithme d'estimation hiérarchique du modèle, facilitant notamment l'emploi d'un modèle de caméra perspectif et la sélection automatique du nombre de modes par validation croisée. Des résultats expérimentaux sur des vidéos variées montrent que l'approche proposée reconstruit des déformations plausibles de la surface observée, permettant l'augmentation 3D de surfaces déformables sur une vidéo. Ces travaux ont été publiés dans (Bartoli et al., 2008).

---

---

### 5.1 Introduction

L'estimation de la structure et du mouvement à partir d'une vidéo prise par une seule caméra est un problème largement étudié en vision par ordinateur. Pour une scène statique (rigide), les rayons de vue associés au même point 3D observé par la caméra à des positions différentes s'intersectent dans l'espace. Cette propriété permet de définir des contraintes fortes sur la reconstruction. La reconstruction 3D de scènes rigides est en général un problème bien posé. Pour une scène dynamique, l'hypothèse que les rayons de vue s'intersectent n'est plus valide : évaluer la surface 3D, ses déformations et le mouvement de la caméra à partir d'une vidéo est dans la plupart des cas un problème sous contraint. En général, seule une approximation des déformations de la surface peut être estimée, à l'exception de certaines configurations bien contraintes. Par exemple, il a été démontré récemment dans (Taddei and Bartoli, 2008) que des déformations très spécifiques d'une page d'un livre peuvent être reconstruites de manière exacte.

Afin de garantir la reconstruction de déformations plausibles, il est indispensable de limiter l'espace des déformations admissibles en incorporant, au processus de reconstruction, des informations *a priori* sur la scène observée. Elles peuvent être spécifiques à une surface donnée, par un exemple un modèle de visage ou de papier, ou plus générique, comme par exemple des déformations lisses. La

figure 5.1 répertorie des exemples d'informations *a priori* utilisées dans la littérature pour la reconstruction 3D d'objets déformables, elles sont classées en fonction de leur généralité.

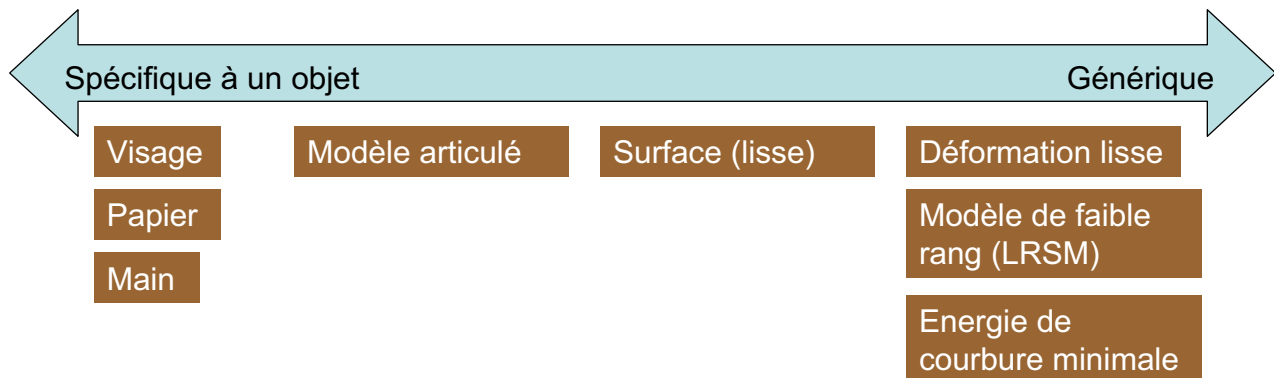


FIG. 5.1 – Exemples d'informations *a priori* utilisées dans la littérature pour la reconstruction 3D d'objets déformables. Elles vont du très spécifique (comme par exemple un modèle de visage) au générique (comme par exemple des déformations lisses).

L'approximation des déformations d'une surface par une combinaison linéaire de modes de déformation est l'une des représentations les plus couramment utilisées dans la littérature en raison de sa capacité à modéliser une grande variété de déformations. La principale hypothèse est que la scène dynamique observée est constituée d'un seul objet (pour nous une surface) telles que les déformations en chaque point soient « consistantes » avec celles des autres points.

Les modes de déformation sont soit appris sur une base de données ; on parle alors de modèle pré-appris ou morphable 3D ; soit directement estimés sur les données courantes ; on parle alors de modèle non appris ou de faible rang. Les modèles pré-appris sont spécifiques à un type de surface. Ils sont suffisamment contraints pour permettre la reconstruction 3D des déformations à partir d'une seule image. En contrepartie, la structure présente dans les images est supposée *a priori* connue. Ce type d'approche est notamment utilisé pour la reconstruction 3D de visage (Blanz and Vetter, 1999) et de surface planes (Salzmann et al., 2005). Les modèles non appris sont plus génériques et présentent l'avantage de s'adapter à la structure présente dans les images. Nous nous intéressons principalement à ces méthodes dans ce chapitre.

Les algorithmes de reconstruction utilisant le modèle de faible rang, par exemple (Aanaes and Kahl, 2002; Brand, 2001, 2005; Bregler et al., 2000; Del Bue et al., 2006; Olsen and Bartoli, 2008; Torresani et al., 2008; Xiao et al., 2004; Xiao and Kanade, 2005), ont montré leur efficacité. La principale différence avec l'algorithme que nous proposons se situe au niveau de la représentation du modèle. La plupart des méthodes existantes traitent l'ensemble des modes équitablement. Il en résulte des ambiguïtés puisque chaque mode de déformation peut être remplacé par une combinaison linéaire des autres modes. Nous proposons au contraire de les ordonner par importance en terme d'amplitude de déformation capturée dans les images. Notre approche est semblable à l'ACP<sup>1</sup> d'un jeu de données pour laquelle les modes de déformation sont ordonnés en terme de variance capturée. La principale différence est que les modes de déformations sont estimés par minimisation de l'erreur de reprojection.

Cette représentation permet de lever certaines ambiguïtés et introduit une estimation *hiérarchique*

<sup>1</sup>Analyse en Composantes principales.

du modèle de faible rang. La sélection automatique du nombre de modes de déformation est une problématique qui est très peu étudiée. Nous proposons d'utiliser la *validation croisée* pour faire ce choix. Notons que l'algorithme proposé n'est pas *incrémental* ; des modes ne peuvent pas être ajoutés au fur et à mesure que les images arrivent.

En résumé, nous proposons un nouvel algorithme de reconstruction 3D basé sur le modèle de faible rang. Ce dernier gère les données manquantes, utilise un modèle de caméra perspectif, sélectionne automatiquement le nombre de modes de déformation et utilise des informations *a priori* complémentaires favorisant la reconstruction de formes 3D plausibles. Il permet en outre l'augmentation 3D de surfaces déformables sur une vidéo.

**Plan.** Les modèles représentant les déformations d'une surface par une combinaison linéaire de modes sont décrits en §5.2. Nous verrons notamment que le modèle de faible rang sous sa forme explicite est la représentation la plus générique. Ensuite, les algorithmes estimant la forme explicite du modèle de faible rang sont passés en revue en §5.3. Nous proposons notre nouvelle approche basée sur le concept de modèle de faible rang hiérarchique en §5.4. Les résultats expérimentaux sont présentés en §5.5. Pour finir, nous donnons nos conclusions et discutons des travaux futurs en §5.6.

## 5.2 Approximation des déformations par combinaison linéaire de modes de déformation

Nous nous intéressons ici aux modèles représentant les déformations d'une surface par une combinaison linéaire de modes de déformation. Cette représentation présente les avantages d'être flexible et de s'adapter à une grande variété de surfaces. Considérons dans un premier temps que la nature de la surface observée soit *a priori* connue. Il est alors possible d'apprendre au préalable les modes de déformation par ACP d'une base de données suffisamment représentative. On parle alors de modèles morphables 3D ou modèles pré-appris. Des travaux ont récemment été proposés pour estimer les modes de déformation sans connaissance préalable de la scène. On parle alors de modèle de faible rang (LRSM<sup>2</sup>) ou modèle non appris, pour lequel les modes sont estimés directement sur les données courantes.

En résumé, il existe deux grandes approches permettant de reconstruire les déformations d'une surface représentée par une combinaison linéaire de modes de déformation. La première consiste à apprendre le modèle au préalable tandis que la deuxième estime les modes sur les données courantes. Les modèles pré-appris sont dédiés à un type de surface, tandis que les modèles non appris s'adaptent aux données. Les modèles pré-appris sont plus stables et mieux posés ce qui permet de les estimer à partir d'une seule image. Ils sont en contrepartie moins génériques.

### 5.2.1 Modèles pré-appris

L'apprentissage consiste à estimer hors ligne les modes de déformation à partir de données préalablement recalées. Cette étape est équivalente à l'ajustement d'un modèle non appris à des données. Elle est cependant effectuée avec des données qui facilitent le processus, comme des visages numérisés en 3D (Blanz and Vetter, 1999) ou bien des déformations synthétiques d'une surface plane (Salzmann et al., 2005).

---

<sup>2</sup>"Low-Rank Shape Model" en anglais.

### 5.2.1.1 Les modèles morphables 3D

La position d'un point 3D dans les modèles morphables 3D (3DMMs<sup>3</sup>) est issue d'une combinaison linéaire de  $l$  modes de déformation  $\{\mathbf{B}_{k,j}\}_{k=1}^l$  appris, avec des coefficients de forme  $\{a_k\}_{k=1}^l$ . Le point image  $\mathbf{s}_j$  est obtenu via l'opérateur de projection  $\Pi$  :

$$\mathbf{s}_j \stackrel{\text{def}}{=} \Pi \left( \sum_{k=1}^l a_k \mathbf{B}_{k,j} \right).$$

Choisir  $l = 1$  correspond à une surface rigide.

### 5.2.1.2 Exemples de modèles morphables 3D

**Modèle de surfaces planes.** Dans (Salzmann et al., 2005), un modèle approximant les déformations de surfaces telles que le tissu ou bien encore une feuille de papier est proposé. Il est construit par apprentissage des modes de déformation sur un jeu de données généré synthétiquement, voir §2.4.2 pour plus de détails. Le principal inconvénient du modèle obtenu est que la propriété d'inextensibilité est perdue au cours du processus d'apprentissage, cette dernière étant non-linéaire. Des modes de déformation encodant la dilatation ou la contraction du maillage rendent la reconstruction parfois ambiguë.

Afin de palier à cet inconvénient, la contrainte non-linéaire d'inextensibilité est réinjectée dans la fonction de coût via une pénalité  $\mathcal{E}_n$ . Elle est appliquée aux longueurs entre les sommets du maillage :

$$\mathcal{E}_n \stackrel{\text{def}}{=} \sum_{j=1}^m \sum_{\mathbf{s}_p \in \mathbf{N}(\mathbf{s}_j)} \left( \|\mathbf{s}_j - \mathbf{s}_p\|^2 - L_{j,p}^2 \right)^2, \quad (5.1)$$

avec  $\mathbf{N}(\mathbf{s}_j)$  l'ensemble des sommets voisins de  $\mathbf{s}_j$  et  $L_{j,p}$  la distance initiale, c'est-à-dire avant déformation, entre les sommets  $\mathbf{s}_j$  et  $\mathbf{s}_p$ . La représentation des surfaces inextensibles par cette pénalité est une approximation.

**Modèle de visage.** Dans (Blanz and Vetter, 1999), un modèle de visage est proposé. Il est issu d'une ACP appliquée à des jeux (texture et forme) de données réelles provenant d'un scanner 3D. Plus de détails sont donnés en §2.4.2. Ce modèle n'est pas suffisamment contraint pour garantir que la surface reconstruite est valide. Les variations de forme et de texture sont alors strictement limitées à l'espace engendré par les données d'apprentissage. Des pénalités supplémentaires sont insérées dans la fonction de coût. Elles s'appliquent aux *coefficients de texture* et de forme  $\{a_k^T\}_{k=1}^l$  et  $\{a_k^S\}_{k=1}^l$  associés aux *modes d'apparence* et de déformation :

$$\sum_{k=1}^{l_T} \left( \frac{a_k^T}{\sigma_k^T} \right)^2 \quad \text{et} \quad \sum_{k=1}^{l_S} \left( \frac{a_k^S}{\sigma_k^S} \right)^2,$$

où  $\{\sigma_k^T\}_{k=1}^l$  et  $\{\sigma_k^S\}_{k=1}^l$  sont les valeurs propres des matrices de covariance utilisées pour apprendre les modèles de texture et de forme. Elles traduisent la représentativité des données d'apprentissage suivant les modes d'apparence et de déformation.

Au final, la fonction de coût combinée comprend un terme de données et les contraintes décrites ci-dessus. Il est admis que la base d'apprentissage est suffisamment représentative de la classe visage ; cette hypothèse peut difficilement être totalement garantie.

<sup>3</sup>"3D Morphable Models" en anglais.

## 5.2.2 Modèles non appris

Contrairement aux modèles appris, les modèles non appris doivent découvrir la structure de la scène dans le jeu de données courant. Ils sont utiles lorsque la surface observée n'est pas *a priori* connue.

### 5.2.2.1 Le modèle de faible rang explicite

Le modèle de faible rang explicite LRSM s'écrit comme les modèles morphables mais nécessite l'introduction d'un indice de vue  $i$  car il ne peut être ajusté à une seule image. Un point image  $\mathbf{s}_{i,j}$  est donné par :

$$\mathbf{s}_{i,j} \stackrel{\text{def}}{=} \Pi_i \left( \sum_{k=1}^l a_{i,k} \mathbf{B}_{k,j} \right). \quad (5.2)$$

Chaque mode permet à un point 3D de se déplacer le long d'une certaine direction qui est fonction du point  $j$  avec une amplitude dépendant à la fois du point  $j$  et de la vue  $i$ .

L'estimation du modèle LRSM sur les données courantes le rend plus flexible et générique. En contrepartie, il est moins contraint que les modèles pré-appris. Évaluer les paramètres du modèle LRSM est un problème complexe pour lequel de nombreuses solutions ont été proposées ; elles sont décrites en §5.3. La terminologie « faible rang » provient du rang de la matrice de mesure dans le formalisme de factorisation présenté en §5.3.1 ; le rang étant très inférieur au nombre de points  $m$  constituant le modèle et au nombre d'images  $n$  sur lesquelles il est estimé :  $l \ll \min(n, m)$ .

### 5.2.2.2 Le modèle de faible rang implicite

La forme implicite du modèle de faible rang est obtenue à partir de sa forme explicite donnée par l'équation (5.2). Si un modèle de projection *orthographique*<sup>4</sup> est utilisé, l'équation (5.2) peut être réécrite sous la forme :

$$\mathbf{s}_{i,j} = \bar{\mathbf{R}}_i \left( \sum_{k=1}^l a_{i,k} \mathbf{B}_{k,j} \right) + \bar{\mathbf{t}}_i. \quad (5.3)$$

En rassemblant les coefficients de forme et la partie rotationnelle des projections orthographiques dans des matrices composites  $\{\mathbf{W}_i\}_{i=1}^n$  et les modes de déformation dans les vecteurs  $\{\mathbf{U}_j\}_{j=1}^m$ , on obtient :

$$\mathbf{s}_{i,j} = \begin{pmatrix} a_{i,1} \bar{\mathbf{R}}_i & \cdots & a_{i,l} \bar{\mathbf{R}}_i \end{pmatrix} \begin{pmatrix} \mathbf{B}_{1,j} \\ \vdots \\ \mathbf{B}_{l,j} \end{pmatrix} + \bar{\mathbf{t}}_i = \mathbf{W}_i \mathbf{U}_j + \bar{\mathbf{t}}_i, \quad (5.4)$$

où  $\mathbf{W}_i$  et  $\mathbf{U}_j$  ont pour taille respective  $(2 \times r)$  et  $(r \times 1)$ , avec  $r = 3l$  le rang de la matrice de mesure dans le formalisme de factorisation décrit en §5.3.1.

La forme implicite du LRSM est donnée par :

$$\mathbf{q}_{i,j} \stackrel{\text{def}}{=} \mathbf{J}_i \mathbf{Z}_j + \bar{\mathbf{t}}_i \quad \text{avec} \quad \mathbf{J}_i \stackrel{\text{def}}{=} \mathbf{W}_i \mathbf{Q}^{-1} \quad \text{et} \quad \mathbf{Z}_j \stackrel{\text{def}}{=} \mathbf{Q} \mathbf{U}_j.$$

<sup>4</sup>Un modèle de projection orthographique est un modèle de caméra affine pour lequel les paramètres  $\alpha_x$ ,  $\alpha_y$  et  $\tau$ , de l'équation (1.1), sont fixés à 1.

Les matrices  $J^T = \begin{pmatrix} J_1^T & \cdots & J_n^T \end{pmatrix}$  et  $Z = \begin{pmatrix} Z_1 & \cdots & Z_m \end{pmatrix}$  sont respectivement les matrices de mouvement et de forme implicite. La matrice  $Q_{(r \times r)}$  est appelée *matrice de correction* ; elle représente la transformation correctrice qui permet de passer de la forme implicite à la forme explicite du modèle. La principale différence entre ces deux formes de LRSM est que la matrice de forme implicite  $J$  n'a pas la structure répétitive par blocs caractérisant la matrice de forme explicite  $W$ . De plus la forme explicite est au moins trinéaire tandis que la forme implicite est au moins bilinéaire. Estimer les paramètres de ce modèle constitue la première étape de l'approche stratifiée utilisée par de nombreux auteurs et décrite ci-dessous.

Notons que la structure par blocs de la matrice  $W$  n'est pas requise si l'objectif est de déterminer le mouvement des points 2D pour la prédiction des données manquantes ou le filtrage des données aberrantes. La forme implicite du modèle de faible rang est alors suffisante (Olsen and Bartoli, 2008). En revanche, si l'objectif est de reconstruire la scène observée (comme c'est le cas dans ce chapitre), la forme explicite du modèle de faible rang doit être évaluée.

### 5.2.2.3 Les informations *a priori*

Il a été montré par plusieurs auteurs, comme par exemple dans (Torresani et al., 2008), que le modèle LRSM était d'autant mieux posé que des informations *a priori* étaient utilisées. La principale raison est que le modèle de faible rang est empirique et par conséquent très sensible au nombre  $l$  de modes de déformation. Si  $l$  est trop grand alors les degrés de liberté supplémentaires ne sont pas contraints par les données image. Le modèle a alors tendance à modéliser les erreurs introduites par sa propre approximation de la physique de la surface observée. Au contraire, si  $l$  est trop petit alors le modèle ne peut plus représenter l'ensemble des déformations de la surface. Pour pouvoir résoudre les problèmes décrits ci-dessus, l'ajout d'informations supplémentaires dites *a priori* est requis. Ces dernières permettent de mieux contraindre le modèle de faible rang et le rendent moins sensible au nombre de modes. Elles limitent l'espace des déformations admissibles en les contraignant temporellement et spatialement.

Une grande variété de modèles d'informations *a priori* est proposée dans la littérature. Elles se doivent d'être le plus génériques possible, c'est-à-dire non spécifiques à un objet ou une classe d'objets. Un exemple d'information *a priori* spécifique est l'annulation de la *courbure Gaussienne* (Perriollat and Bartoli, 2007) qui modélise bien les propriétés d'une feuille de papier mais n'est pas valide pour un visage ou un tissu quelconque. De manière générale, il existe plusieurs classes d'informations *a priori* :

- ▷ Celles contraignant le déplacement des caméras.
- ▷ Celles imposant des contraintes temporelles sur les déformations de la surface.
- ▷ Celles contrôlant les déformations admissibles du point de vue spatial.

Nous rapportons ci-dessous quelques exemples d'informations *a priori* « génériques » proposées dans la littérature. Pour la plupart des vidéos, le déplacement de la caméra est régulier. Une pénalité, se basant sur cette observation, est proposée dans (Olsen and Bartoli, 2008). Elle impose un déplacement lisse des caméras pour la forme implicite du modèle de faible rang :  $\sum_{i=1}^{n-1} \|J_i - J_{i+1}\|^2$ . Notons que les coefficients de forme associés aux modes de déformation sont également encapsulés dans chaque matrice de mouvement implicite  $J_i$  ; cette pénalité leur impose par conséquent également des variations lisses.

Dans (Del Bue et al., 2004), une contrainte sur les variations temporelles de la profondeur de la

surface observée est proposée :  $\sum_{i=2}^n \sum_{j=1}^m \|S_{i-1,j}^z - S_{i,j}^z\|^2$ , tandis que cette contrainte est étendue dans (Aanæs and Kahl, 2002) à toutes les coordonnées  $\sum_{i=2}^n \sum_{j=1}^m \|S_{i-1,j} - S_{i,j}\|^2$ . Ces informations complémentaires sont valides si la surface observée ne se déforme pas trop entre deux images.

Les variations temporelles sont modélisées dans (Torresani et al., 2008) par un modèle dynamique linéaire appliqué aux coefficients de forme  $\{\mathbf{a}_i\}_{i=1}^n : \mathbf{a}_i = \Phi \mathbf{a}_{i-1}$ , avec  $\Phi_{l \times l}$  une matrice de transition et  $\mathbf{a}_i^T = (a_{i,1} \ \dots \ a_{i,l})$ . Cette matrice est estimée au cours du processus d'optimisation avec les paramètres du modèle. Les auteurs supposent également que les coefficients de forme peuvent être représentés par une distribution Gaussienne de moyenne nulle. Les coefficients situés en dehors de la distribution deviennent marginaux, limitant ainsi l'espace des déformations admissibles.

Récemment, il a été proposé dans (Del Bue, 2008) d'utiliser la composante rigide associée à une classe donnée (visage, feuille de papier, *etc.*) pour contraindre les déformations à être physiquement admissibles. Elle est estimée au préalable sur une vidéo ne présentant que des mouvements rigides d'un des éléments de la classe. La pénalité introduite impose à la forme globale de la surface à être le plus semblable possible à la composante rigide évaluée au préalable. Cette contrainte permet une estimation raisonnable des déformations de la surface observée même si la vidéo présente des cas dégénérés comme par exemple une personne parlant devant une caméra sans mouvement de tête. Cette information est bien générique au sens qu'elle peut être appliquée à différents types de surfaces. Cependant, elle nécessite la connaissance *a priori* de la classe auquel appartient la surface observée. Cette approche peut être vue comme un compromis entre les modèles non appris et ceux pré-appris.

D'autres informations *a priori* peuvent être trouvées dans la littérature, comme celle contraignant l'orientation des sommets d'un maillage régulier à ne pas changer démesurément entre deux images consécutives (Salzmann et al., 2007a). Elle est réécrite sous forme de contraintes convexes permettant son optimisation par une approche de type "Second-Order Cone Programming".

## 5.3 Estimation de la forme explicite du modèle de faible rang

Nous présentons dans cette section les différentes méthodes permettant d'estimer le modèle de faible rang sous sa forme explicite. La plupart d'entre elles minimisent l'erreur de reprojection comme terme de données. Elles se différencient principalement au niveau du processus de minimisation.

### 5.3.1 L'approche stratifiée

La plupart des algorithmes suivent une approche *stratifiée* pour reconstruire la forme explicite du LRSM. Elle s'articule habituellement autour de 3 étapes :

**Etape 1 : factorisation de la matrice de mesure.** Cette étape est l'extension de la factorisation rigide initialement proposée dans (Tomasi and Kanade, 1992). Soit la matrice de mesure  $\mathcal{M}$  rassemblant l'ensemble des points 2D mesurés au cours de la vidéo. Ces derniers sont au préalable translattés sur leur centre de gravité. La forme implicite du modèle de faible rang est obtenue en factorisant la matrice  $\mathcal{M}$ . La plupart des algorithmes supposent que le rang  $r$  est connu et qu'il n'y a pas de données manquantes et erronées. Sous ces hypothèses, une solution peut être obtenue par une SVD<sup>5</sup> de

<sup>5</sup>"Singular Value Decomposition".

$\mathcal{M}$  tronquée au rang  $r$  :

$$\mathcal{M}_{2n \times m} = \begin{pmatrix} \mathbf{q}_{1,1} & \cdots & \mathbf{q}_{1,m} \\ \vdots & & \vdots \\ \mathbf{q}_{n,1} & \cdots & \mathbf{q}_{n,m} \end{pmatrix} = \mathbf{J}_{2n \times r} \mathbf{Z}_{r \times m} + \Psi_{2n \times m}. \quad (5.5)$$

En raison du bruit sur les données et de l'écart entre la physique de la surface observée et le modèle de faible rang implicite, l'égalité  $\mathcal{M} = \mathbf{J}\mathbf{Z}$  ne peut jamais être vérifiée. Les erreurs résiduelles sont encapsulées dans la matrice de « bruit »  $\Psi$  (de taille  $(2n \times m)$ ). La figure 5.2 illustre le principe de factorisation de la matrice de mesure en une matrice de mouvement et une matrice de forme implicite.

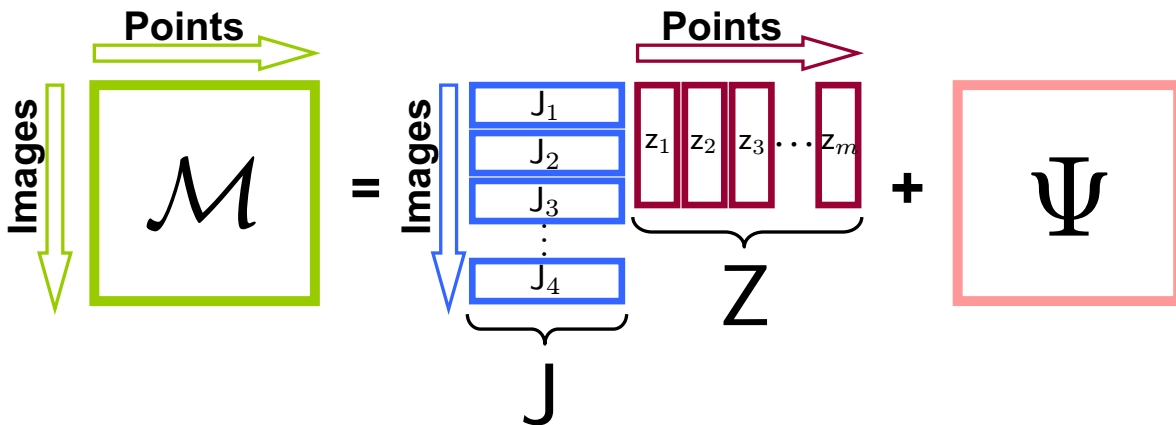


FIG. 5.2 – **Factorisation de la matrice de mesure.** La matrice de mesure est décomposée en une matrice de mouvement implicite  $\mathbf{J}$  et une matrice de forme implicite  $\mathbf{Z}$ . La matrice de « bruit »  $\Psi$  représente ce que le modèle ne peut pas expliquer dans les données.

De nombreuses améliorations ont été proposées afin d'étendre cette approche à des cas plus proches de la réalité pour lesquels la présence de données manquantes et / ou erronées est inévitable. Dans (Olsen and Bartoli, 2008) la forme implicite du modèle de faible rang est estimée à partir de contraintes de fermeture issues de sous blocs de la matrice de mesure ne présentant pas de données manquantes. Les différentes contraintes introduites par ces sous blocs sont combinées afin d'estimer la matrice de mouvement implicite par moindres carrés linéaires. La matrice de forme est quant à elle obtenue par triangulation des  $\{\mathbf{Z}_j\}_{j=1}^m$ . Notons que la factorisation de la matrice de mesure est étendue à un modèle de caméra perspectif dans (Vidal and Abretské, 2006; Xiao and Kanade, 2005).

**Etape 2 : estimation de la transformation correctrice.** La deuxième étape de l'approche stratifiée est d'estimer la transformation correctrice  $\mathbf{Q}$  permettant de passer de la forme implicite à la forme explicite du modèle de faible rang :  $\mathbf{W} = \mathbf{J}\mathbf{Q}$  et  $\mathbf{U} = \mathbf{Q}^{-1}\mathbf{Z}$ . Cette transformation doit imposer la structure répétitive par blocs de la matrice de mouvement : cette étape constitue la principale difficulté et est sujette à de nombreux travaux de recherche comme par exemple (Brand, 2005; Brand and Bhotika, 2001; Bregler et al., 2000; Xiao et al., 2004). Nous présentons plus en détails en §5.3.1.1 et §5.3.1.2 deux approches estimant cette matrice de correction. Une fois les matrices de mouvement et de forme évaluées, les coefficients de forme  $\{a_{i,k}\}_{i,k=1}^{n,l}$  et les poses  $\{\bar{\mathbf{R}}_i\}_{i=1}^n$  sont extraits de la matrice de mouvement explicite  $\mathbf{W}$ .



**Étape 3 : raffinement par ajustement de faisceaux.** Cette dernière étape est optionnelle. L'erreur de reprojection est minimisée sur tous les points  $j$  et toutes les vues  $i$  par l'algorithme Levenberg-Marquardt. Cette étape est semblable à celle d'ajustement de faisceaux des algorithmes de reconstruction 3D de scènes rigides que nous avons présenté en §1.4.3. Le critère suivant est minimisé :

$$\min_{\{\Pi_i\}_{i=1}^n, \{a_{i,k}\}_{i,k=1}^{n,l}, \{\mathbf{B}_{k,j}\}_{k,j=1}^{l,m}} \sum_{i=1}^n \sum_{j=1}^m v_{i,j} d^2(\mathbf{q}_{i,j}, \Pi_i(\mathbf{S}_{i,j})), \quad (5.6)$$

avec  $v_{i,j}$  un élément de la carte de visibilité  $\mathbf{V}$ , celle-ci indiquant si le point 3D  $j$  est visible dans l'image  $i$ .

C'est un problème mal posé puisque chaque mode de déformation peut être remplacé par une combinaison linéaire des autres modes sans pour autant modifier la prédiction d'un point 3D  $\mathbf{S}_{i,j}$  et par conséquent sans modifier sa projection  $s_{i,j}$ . Des informations *a priori* complémentaires, comme par exemple celles présentées en §5.2.2.3, peuvent être incorporées dans la fonction de coût afin de régulariser la solution.

### 5.3.1.1 L'approche de Bregler *et al.*

Cette approche, proposée dans (Bregler et al., 2000), est basée sur l'hypothèse que toutes les composantes du modèle de faible rang dépendant des images c'est-à-dire les coefficients de forme et les paramètres des caméras, sont entièrement contenus dans la matrice de mouvement implicite  $\mathbf{J}$ . L'extraction des composantes de mouvement est alors effectuée par SVD au rang 1 des  $n$  sous blocs  $\{\check{\mathbf{J}}_i\}_{i=1}^n$  obtenus en réordonnant les matrices de mouvement implicites  $\{\mathbf{J}_i\}_{i=1}^n$  :

$$\check{\mathbf{J}}_i = \begin{pmatrix} a_{i,1} \mathbf{r}_i^T \\ \vdots \\ a_{i,l} \mathbf{r}_i^T \end{pmatrix} + \Lambda_{(l \times 6)} = \begin{pmatrix} a_{i,1} \\ \vdots \\ a_{i,l} \end{pmatrix} \begin{pmatrix} r_{i,1} & r_{i,2} & r_{i,3} & r_{i,4} & r_{i,5} & r_{i,6} \end{pmatrix} + \Lambda_{(l \times 6)}. \quad (5.7)$$

où les  $\{r_{i,p}\}_{p=1}^6$  sont les éléments de la matrice  $\bar{\mathbf{R}}$  et  $\Lambda_{(l \times 6)}$  une matrice de « bruit ». Une correction est ensuite appliquée aux matrices de rotation afin d'imposer leur orthonormalité. L'hypothèse de Bregler *et al.* n'est cependant pas valide en pratique car les coefficients de forme ainsi que les matrices de rotation doivent être extraits de la matrice de mouvement explicite  $\mathbf{W}$  obtenue après correction de la forme implicite du modèle.

### 5.3.1.2 L'approche de Xiao *et al.*

Elle consiste à corriger la forme implicite du modèle en utilisant les contraintes relatives aux matrices de rotation mais également des contraintes sur les modes de déformation. En effet, il est montré dans (Xiao et al., 2004) que les contraintes d'orthonormalité imposées sur les matrices de rotation ne sont pas suffisantes pour rendre le problème bien posé : il existe plusieurs couples formes 3D - caméras pouvant expliquer les données. Afin de résoudre cette ambiguïté des contraintes sur les modes de déformation, appelées *contraintes de base*, doivent également être imposées. Elles permettent, en complément des contraintes sur les matrices de rotation, d'aboutir à une solution analytique pour passer de la forme implicite à la forme explicite du modèle de faible rang.

Les modes de déformation ne sont effectivement pas uniques puisque pour toute transformation linéaire non-singulière, un nouvel ensemble de modes peut être généré. Cependant, si il existe  $l$  images

pour lesquelles les modes de déformation sont indépendants, alors ces derniers peuvent être déterminés de manière unique. Pour ces images, la forme 3D associée est expliquée par un seul mode de déformation. Ceci revient à imposer la structure suivante à la matrice de mesure :

$$\mathcal{M} = \begin{pmatrix} \bar{R}_1 & 0 & \cdots & 0 \\ 0 & \bar{R}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \bar{R}_l \\ \hline a_{l+1,1}\bar{R}_{l+1} & a_{l+1,2}\bar{R}_{l+1} & \cdots & a_{l+1,l}\bar{R}_{l+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\bar{R}_n & a_{n,2}\bar{R}_n & \cdots & a_{n,l}\bar{R}_n \end{pmatrix} \begin{pmatrix} B_1 \\ \vdots \\ B_l \end{pmatrix} + \Psi, \quad (5.8)$$

avec  $B_i = \begin{pmatrix} B_{i,1} & \cdots & B_{i,m} \end{pmatrix}$ . Elle est sur cette exemple réarrangée pour que les  $l$  images associées aux modes de déformation indépendants correspondent à ses  $l$  premières lignes.

Les contraintes sur les modes sont déduites des relations suivantes :

$$\begin{aligned} a_{i,i} &= 1 & i &= [1, \dots, l] \\ a_{i,k} &= 0 & i, k &= [1, \dots, l], i \neq k. \end{aligned}$$

Les performances de cette méthode se dégradent fortement lorsque les  $l$  formes indépendantes sont mal sélectionnées comme démontré dans (Brand, 2005). Notons que même si une solution unique est estimée celle-ci diffère en fonction du choix des  $l$  modes. Leur extraction s'avère être complexe, de manière unique, sur des vidéos réelles présentant une surface déformable.

**Discussion.** La grande difficulté relative aux approches stratifiées est l'estimation de la transformation corrective imposant la structure répétitive par blocs de la matrice de mouvement requise par la forme explicite du modèle de faible rang. Les approches existantes n'offrent pour l'instant pas de solution entièrement satisfaisante : les paramètres de mouvement sont extraits à partir de la forme implicite du modèle de faible rang dans (Bregler et al., 2000). Dans (Brand and Bhotika, 2001) la forme implicite est corrigée au préalable par des contraintes d'orthonormalité des matrices de rotation. Enfin, dans (Xiao et al., 2004), il est démontré qu'une solution estimée uniquement à partir des contraintes sur les matrices de rotation peut être dégénérée et non unique. Ils proposent en complément des contraintes de base en faisant l'hypothèse qu'il existe  $l$  images pour lesquelles les modes de déformation sont indépendants. L'existence de telles images n'est pas garantie sur des cas réels. De nombreux travaux proposent d'estimer directement la forme explicite du modèle de faible rang sans passer par sa forme implicite, évitant ainsi la difficulté introduite par l'estimation de la transformation corrective. Elles sont décrites ci-dessous.

### 5.3.2 Estimation directe de la forme explicite

Les algorithmes décrits ci-dessous estiment directement la forme explicite du modèle. L'étape complexe assurant le passage de la forme implicite à la forme explicite du modèle de faible rang est ainsi évitée.

Dans (Torresani et al., 2001) un schéma d'optimisation minimisant alternativement, au sens des moindres carrés, les trois sous classes constituant les paramètres du modèle  $\{\Pi_i\}_{i=1}^n$ ,  $\{a_{i,k}\}_{i,k=1}^{n,l}$

et  $\{\mathbf{B}_{k,j}\}_{k,j=1}^{l,m}$ , est défini. Lors de l'estimation de l'une des classes les deux autres sont laissées constantes. Cette procédure est connue sous le nom alternation et possède l'avantage de garantir la convergence vers un minimum local sans la complexité des approches totalement non-linéaires. Une initialisation appropriée des matrices de projection peut être obtenue par factorisation rigide (voir §1.4). Les coefficients de forme sont quant à eux initialisés aléatoirement permettant ainsi une première évaluation des modes de déformation. Notons que la mise à jour de chaque groupe de paramètres possède une forme analytique, excepté pour les matrices de projection. Cet algorithme présente l'inconvénient de minimiser des distances algébriques. Ce mécanisme d'optimisation converge lentement comme rapporté dans (Buchanan and Fitzgibbon, 2005). Ce schéma d'optimisation est reformulé, dans (Torresani et al., 2008), sous forme probabiliste en incorporant des informations *a priori* Gaussienne sur les coefficients de formes. L'algorithme EM<sup>6</sup> est alors utilisé pour l'optimisation.

L'algorithme Levenberg-Marquardt est souvent privilégié pour minimiser l'équation (5.6). Il présente de bien meilleures performances de convergence comme l'illustrent les résultats obtenus dans (Buchanan and Fitzgibbon, 2005). L'algorithme Levenberg-Marquardt est initialisé dans (Buchanan and Fitzgibbon, 2005) en tirant aléatoirement un grand nombre de conditions initiales (de l'ordre de 1000). La solution donnant l'erreur la plus faible en sortie est conservée. Ceci s'avère être extrêmement coûteux en temps de calcul. Des initialisations plus proches du minimum global ont été proposées dans la littérature.

Dans (Del Bue et al., 2006), l'hypothèse raisonnable que certaines régions de la scène observée se déforment au cours de la vidéo tandis que d'autres restent sensiblement rigides<sup>7</sup> est exploitée. Ces dernières peuvent être automatiquement segmentées en mesurant la variance d'un point par rapport à son positionnement moyen (obtenu par factorisation de la matrice de mesure par l'algorithme proposé dans (Tomasi and Kanade, 1992)). Une fois les régions rigides segmentées, un algorithme de reconstruction rigide classique (Hartley and Zisserman, 2003) permet de calibrer les caméras et d'estimer leur pose ainsi que la forme moyenne à partir de ces régions. Les modes de déformation et les coefficients de forme sont quant à eux fixés à de faibles valeurs de manière aléatoire. Ils prétendent que cette initialisation *ad hoc* des modes ne dégrade pas le bassin de convergence de l'algorithme Levenberg-Marquardt.

Dans (Aanæs and Kahl, 2002) le modèle de faible rang est vu comme un ensemble de modes ordonnés en fonction de la variance des données qu'ils capturent. Le premier mode de déformation ainsi que les caméras sont au préalable évalués par des algorithmes de reconstruction de scène rigide (Tomasi and Kanade, 1992). Les autres modes de déformation sont issus d'une ACP de la variance des points 3D. Les auteurs montrent que cette dernière peut être directement estimée à partir de la matrice de mesure si les matrices de projection sont connues. Le tout est ensuite raffiné par l'algorithme Levenberg-Marquardt. L'ACP engendre des modes de déformation ordonnés en fonction des valeurs propres, c'est-à-dire en fonction de leur capacité à expliquer la dispersion des données. Notons que cette représentation du modèle de faible rang se rapproche de celle utilisée par la suite dans l'approche que nous proposons. Néanmoins, elle diffère au niveau de l'estimation des modes, obtenue avec notre algorithme, par minimisation de l'erreur de reprojction. Cette représentation possède de nombreux avantages : elle réduit notamment les ambiguïtés inhérentes au modèle de faible rang, comme le décrit la §5.4.1.

<sup>6</sup>"Expectation-maximization" en anglais.

<sup>7</sup>Ceci peut-être également vu comme une information *a priori*.

### 5.3.3 Discussion sur les approches existantes

Nous avons répertorié les principales méthodes de reconstruction basées sur le modèle de faible rang dans le tableau 5.1 ainsi que certaines de leurs propriétés :

- ▷ Estimation directe ou stratifiée de la forme explicite.
- ▷ Modèle de caméra perspectif ou affine.
- ▷ Sélection automatique du nombre de modes.
- ▷ Gestion des données manquantes.
- ▷ Modes de déformation ordonnés.

Algorithme	Explicite	Perspectif	Sélection de $l$	Données manquantes	Modes ordonnés
(Aanæs and Kahl, 2002)	●	●	●	×	●
(Brand, 2001)	×	×	×	×	×
(Brand, 2005)	×	×	×	×	×
(Bregler et al., 2000)	×	×	×	×	×
(Buchanan and Fitzgibbon, 2005)	●	●	×	●	×
(Del Bue et al., 2006)	●	●	×	×	×
(Del Bue, 2008)	●	×	×	×	×
(Olsen and Bartoli, 2008)	×	×	●	●	×
(Torresani et al., 2008)	●	×	×	●	×
(Xiao et al., 2004)	×	×	×	×	×
(Xiao and Kanade, 2005)	×	●	×	×	×
Notre approche (§5.4)	●	●	●	●	●

TAB. 5.1 – Comparaison des différentes méthodes de reconstruction 3D basées sur le modèle de faible rang. Un cercle signifie que l’algorithme possède la propriété associée et inversement pour une croix. Pour la colonne « Explicite » un cercle signifie que la forme explicite du modèle est estimée directement, les approches stratifiées sont représentées par une croix.

La plupart des méthodes de reconstruction 3D basées sur le modèle de faible rang utilisent un modèle de caméra orthographique afin de simplifier son estimation. L’emploi d’un modèle perspectif n’a que très peu été étudié dans la littérature, notamment dans (Buchanan and Fitzgibbon, 2005; Del Bue et al., 2006; Xiao and Kanade, 2005). Les algorithmes s’appuyant sur une factorisation de la matrice de mesure par SVD (Brand, 2001, 2005; Bregler et al., 2000; Xiao et al., 2004; Xiao and Kanade, 2005) sont très sensibles aux données manquantes. Une alternative est la méthode de factorisation proposée dans (Olsen and Bartoli, 2008). Le problème de la sélection automatique du nombre de modes a très peu été étudié par les algorithmes précédents. Seuls (Aanæs and Kahl, 2002) et (Olsen and Bartoli, 2008) explorent cette problématique en se basant respectivement sur les critères BIC<sup>8</sup> et GRIC<sup>9</sup>. Nous utilisons dans notre approche un critère basé sur la validation croisée. Enfin, seul (Aanæs and Kahl, 2002) représente le modèle de faible rang comme une combinaison de modes ordonnés en fonction de la variance des données qu’ils capturent. L’approche proposée ordonne également les modes.

<sup>8</sup>"Bayesian Information Criterion".

<sup>9</sup>"Geometric Robust Information Criterion".

## 5.4 Le modèle de faible rang hiérarchique

### 5.4.1 Présentation générale

Dans cette section nous décrivons une nouvelle approche permettant d'estimer le modèle de faible rang. Nous considérons que les modes sont ordonnés par importance en terme d'amplitude de déformation capturée dans les images. Ceci peut être vu comme une information *a priori* incorporée au modèle. Cette représentation présente de nombreux avantages par rapport aux méthodes qui utilisent le modèle de faible rang comme un ensemble de modes non ordonnés. Tout d'abord, les ambiguïtés liées au modèle sont grandement réduites puisque les modes de déformation ne peuvent plus se recombinaison arbitrairement. Ensuite, l'algorithme qui en découle évalue hiérarchiquement les modes : la forme moyenne est dans un premier temps estimée puis des modes sont ajoutés itérativement, la forme explicite du modèle est ainsi directement obtenue. L'étape complexe, décrite en §5.3.1, assurant le passage de la forme implicite à la forme explicite du modèle de faible rang est alors évitée. Enfin, la sélection automatique du nombre de modes de déformation peut facilement être incluse lors de l'estimation. Nous notons également que l'algorithme proposé gère naturellement les données manquantes et permet l'emploi d'un modèle de caméra perspectif.

**Notation.** Dans le reste du chapitre, les différentes instances sont exprimées en coordonnées homogènes introduites en §1.2, le symbole  $\sim$  désignant l'égalité à un facteur près. Nous reformulons le modèle de faible rang comme une forme moyenne  $\{\tilde{\mathbf{M}}_j\}_{j=1}^m$  à laquelle sont ajoutés des modes de déformation :

$$\tilde{\mathbf{S}}_{i,j}^l \stackrel{\text{def}}{=} \tilde{\mathbf{D}}_i \left( \tilde{\mathbf{M}}_j + \sum_{k=1}^l a_{i,k} b_{k,j} \tilde{\mathbf{C}}_{k,j} \right), \quad (5.9)$$

avec  $\tilde{\mathbf{D}}_i$  la pose de la surface observée sur l'image  $i$  et  $\tilde{\mathbf{S}}_{i,j}^l$  un point de l'espace projectif  $\mathbb{P}^3$ . Les modes de déformation sont décomposés en une *direction de déformation* et une *amplitude de déformation* :  $\tilde{\mathbf{B}}_{k,j} = b_{k,j} \tilde{\mathbf{C}}_{k,j}$  avec  $\|\tilde{\mathbf{C}}_{k,j}\| = 1$  et  $\tilde{\mathbf{C}}_{k,j}^\top = \begin{pmatrix} \mathbf{C}_{k,j}^\top & 0 \end{pmatrix}$ . Nous utilisons un modèle de caméra perspectif :  $\tilde{\mathbf{P}}_i = \mathbf{K}_i \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix}$ , supposée fixe puisque la pose relative entre la surface observée et la caméra est représentée par les  $\{\tilde{\mathbf{D}}_i\}_{i=1}^n$ .

La prédiction d'un point image, c'est-à-dire la reprojection d'un point 3D  $\tilde{\mathbf{S}}_{i,j}^l$  prédit par l'équation (5.9) est donnée par :

$$\tilde{\mathbf{s}}_{i,j}^l \sim \tilde{\mathbf{P}}_i \tilde{\mathbf{S}}_{i,j}^l = \tilde{\mathbf{P}}_i \tilde{\mathbf{D}}_i \tilde{\mathbf{M}}_j + \mathbf{P}_i \mathbf{D}_i \sum_{k=1}^l a_{i,k} b_{k,j} \mathbf{C}_{k,j}. \quad (5.10)$$

Nous définissons également les vecteurs :

$$\begin{aligned} \mathbf{a}_l &\stackrel{\text{def}}{=} \begin{pmatrix} a_{1,l} & \cdots & a_{n,l} \end{pmatrix}, \\ \mathbf{b}_l &\stackrel{\text{def}}{=} \begin{pmatrix} b_{l,1} & \cdots & b_{l,m} \end{pmatrix}, \\ \mathbf{C}_l^\top &\stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{C}_{l,1}^\top & \cdots & \mathbf{C}_{l,m}^\top \end{pmatrix}, \\ \mathbf{B}_l^\top &\stackrel{\text{def}}{=} \begin{pmatrix} \mathbf{B}_{l,1}^\top & \cdots & \mathbf{B}_{l,m}^\top \end{pmatrix}. \end{aligned}$$

## 5.4.2 Description du modèle de faible rang hiérarchique

Notre algorithme s'appuie sur la notion de "DEFORMATION" présentée dans (Yezzi and Soatto, 2003). Les auteurs montrent que le mouvement global et les déformations d'une scène peuvent être séparés. Cette décomposition du mouvement n'a pas de sens pour tous les scénarios ; la scène observée doit conserver une certaine « consistance » au cours de ses déformations. Cette définition s'avère être bien adaptée pour les cas que nous souhaitons traiter où une seule surface est observée comme par exemple un visage ou une feuille de papier. Pour des scènes plus complexes composées d'objets multiples, extraire un unique mouvement global n'a pas réellement de sens.

La notion de mouvement global pour une scène déformable est étroitement liée à la notion de forme, ainsi décomposer le mouvement d'une scène en une composante rigide et des déformations est obtenue en estimant sa forme moyenne  $\{\widetilde{\mathbf{M}}_j\}_{j=1}^m$ . Elle constitue la forme à partir de laquelle des déformations minimales sont requises pour aboutir aux  $\{\widetilde{\mathbf{S}}_{i,j}^l\}_{i,j=1}^{n,m}$ .

Une fois la forme moyenne et le mouvement global estimés, les modes de déformation sont ajoutés un à un jusqu'à ce qu'un critère d'arrêt soit satisfait. L'amplitude des déformations capturées décroît au fur et à mesure que des modes sont ajoutés. Il en résulte un ensemble de modes ordonnés.

Cette estimation hiérarchique permet à chaque mode de capturer l'amplitude maximale restante des données qui n'a pas été expliquée par les modes précédents. Le modèle obtenu est semblable à celui issu d'une analyse en composantes principales pour laquelle les modes sont ordonnés en fonction de leur valeur propre, c'est-à-dire en fonction de la variance des données qu'ils capturent.

Notre approche se base sur les relations suivantes, découlant du modèle de faible rang décrit par l'équation (5.9) :

$$\widetilde{\mathbf{S}}_{i,j}^0 = \widetilde{\mathbf{D}}_i \widetilde{\mathbf{M}}_j \quad (5.11)$$

$$\widetilde{\mathbf{S}}_{i,j}^{l+1} = \widetilde{\mathbf{S}}_{i,j}^l + a_{i,l+1} b_{l+1,j} \widetilde{\mathbf{C}}_{l+1,j}. \quad (5.12)$$

En résumé, nous procédons de la manière suivante. Tout d'abord la forme moyenne ainsi que les déplacements  $\{\widetilde{\mathbf{D}}_i\}_{i=1}^n$  recalant globalement la surface déformable au référentiel monde sont estimés par l'équation (5.11) ; ce qui constitue le mode 0 ou la composante rigide du modèle. Ensuite, chaque mode est triangulé<sup>10</sup> itérativement, c'est-à-dire que le  $(l+1)$ <sup>ème</sup> mode de déformation  $\{b_{l+1,j} \widetilde{\mathbf{C}}_{l+1,j}\}_{j=1}^m$  ainsi que les coefficients de forme associés  $\{a_{i,l+1}\}_{i=1}^n$  sont évalués, à partir de l'équation (5.12). Une fonction de coût constituée de l'erreur de reprojection comme terme de données et d'informations *a priori* (décrites ci-dessous) est minimisée à chaque étape. Le tableau 5.2 récapitule le principe d'estimation hiérarchique du modèle de faible rang.

**Informations *a priori*.** En plus de l'information *a priori* implicitement utilisée sur l'ordonnement des modes, nous utilisons explicitement deux informations *a priori* supplémentaires inspirées de (Olsen and Bartoli, 2008). La première contraint les variations temporelles de la surface. Cette hypothèse est valide si la surface observée ne se déforme pas « trop » entre deux images. Elle s'écrit :

$$\mathcal{E}_{as}(\mathbf{a}_{l+1}) \stackrel{\text{def}}{=} \|\Delta \mathbf{a}_{l+1}\|^2,$$

avec  $\Delta$  un opérateur de différence finie approchant la dérivée première. La deuxième contrainte est sur la forme de la surface observée. Elle est basée sur l'observation que des points proches sur la forme

<sup>10</sup>Puisque le mouvement global de la surface est connu à cette étape, nous appelons « triangulation » l'estimation d'un mode.

- ▷ Estimation des composantes rigides du modèle  $\{\widetilde{\mathbf{M}}_j\}_{j=1}^m$  et  $\{\widetilde{\mathbf{D}}_i\}_{i=1}^n$ .
- ▷  $l \leftarrow 0$
- ▷ Début boucle :
  - Ajout d'un mode de déformation : les composantes  $\mathbf{a}_{l+1}$ ,  $\mathbf{b}_{l+1}$ ,  $\mathbf{C}_{l+1}$  sont estimées.
  - $l \leftarrow l + 1$ .
- ▷ Fin boucle si le critère d'arrêt est satisfait.

TAB. 5.2 – Principe de l'estimation hiérarchique du modèle de faible rang.

moyenne le sont aussi après déformation, c'est-à-dire après l'ajout d'un mode. Cette hypothèse est valide dans le cadre de surfaces continues présentant des déformations lisses. Afin de construire cette pénalité, nous estimons la proximité entre chaque point de la forme moyenne :  $\varphi_{j,g} \stackrel{\text{def}}{=} \rho \left( d^2 \left( \widetilde{\mathbf{M}}_j, \widetilde{\mathbf{M}}_g \right) \right)$ , avec  $\rho$  un noyau à support local<sup>11</sup>. La contrainte de surface s'écrit :

$$\mathcal{E}_{bs}(\mathbf{B}_{l+1}) \stackrel{\text{def}}{=} \sum_{j=1}^m \sum_{g=1}^m \varphi_{j,g}^2 \|\mathbf{B}_{l+1,j} - \mathbf{B}_{l+1,g}\|^2 = \|\Omega \mathbf{B}_{l+1}\|^2, \quad (5.13)$$

où  $\Omega$  est une matrice extrêmement creuse ayant  $3m$  colonnes et dont le nombre de lignes est 3 fois le nombre d'éléments non nuls de  $\{\varphi_{j,g}\}_{j,g=1}^{m,m}$ .

**Ambiguïtés.** Si les modes sont estimés simultanément comme c'est le cas pour la plupart des méthodes existantes, il existe alors  $l^2$  degrés d'ambiguïté : chaque mode peut être remplacé par une combinaison linéaire des autres modes. Dans notre approche, le mode  $l + 1$  est conditionné par l'estimation des  $l$  modes précédents, introduisant une seule ambiguïté par mode. En effet, les produits  $\mathbf{a}_l \mathbf{b}_l^T$  (présents dans l'équation (5.12)) entre les coefficients de forme et les amplitudes de déformation, peuvent être redéfinis tel que  $\forall \nu \in \mathbb{R}^* \mathbf{a}_l \mathbf{b}_l^T = (\nu \mathbf{a}_l) (\frac{1}{\nu} \mathbf{b}_l^T)$  :  $\nu$  constitue l'échelle du mode. Il existe également dans tous les cas une transformation Euclidienne indéterminée entre les  $\{\widetilde{\mathbf{D}}_i\}_{i=1}^n$  d'une part et la forme moyenne et les modes d'autre part.

### 5.4.3 Estimation de la forme moyenne

La première étape de notre algorithme est de séparer le mouvement global  $\{\widetilde{\mathbf{D}}_i\}_{i=1}^n$  des déformations de la surface. Nous avons vu en §5.4.2 que ceci peut être réalisé en estimant la forme moyenne  $\{\widetilde{\mathbf{M}}_j\}_{j=1}^m$  c'est-à-dire la composante rigide de la surface. Pour cela, l'erreur de reprojection suivante est minimisée :

$$\min_{\{\widetilde{\mathbf{M}}_j\}_{j=1}^m, \{\widetilde{\mathbf{D}}_i\}_{i=1}^n} \sum_{i=1}^n \sum_{j=1}^m v_{i,j} d^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{P}}_i \widetilde{\mathbf{D}}_i \widetilde{\mathbf{M}}_j). \quad (5.14)$$

C'est un problème d'estimation de la structure à partir du mouvement<sup>12</sup> pour des caméras calibrées. Il est résolu en utilisant des techniques standards décrites en §1.4, comprenant notamment l'ajustement de faisceaux. Notons qu'aucune information *a priori* n'est requise ici puisque la reconstruction 3D de surfaces rigides est habituellement bien posée. Si les paramètres internes des caméras sont inconnus,

<sup>11</sup>Nous utilisons dans nos expériences un noyau Gaussien tronqué.

<sup>12</sup>"Rigid Structure-from-Motion" en anglais.

elles peuvent être calibrées à partir d'une région rigide de la scène observée, par exemple l'arrière plan.

#### 5.4.4 Triangulation d'un mode

Triangler le mode  $l + 1$  revient à minimiser l'erreur suivante :

$$\min_{\mathbf{a}_{l+1}, \mathbf{B}_{l+1}} \sum_{i=1}^n \sum_{j=1}^m d^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{s}}_{i,j}^{l+1}) + \lambda_a \|\Delta \mathbf{a}_{l+1}\|^2 + \lambda_b \|\Omega \mathbf{B}_{l+1}\|^2. \quad (5.15)$$

C'est un problème d'optimisation non-linéaire en raison des produits entre les coefficients de forme, les amplitudes et les directions de déformation, et l'emploi de distances Euclidiennes pour comparer les données images. De manière identique au problème de triangulation rigide, une approximation algébrique des distances peut être utilisée. Néanmoins, le problème reste non-linéaire et complexe à résoudre sous cette forme, les différentes vues et les différents points étant liés. Nous procédons en deux étapes : tout d'abord, les informations *a priori* ne sont pas prises en compte et une estimation initiale du modèle est calculée. Celle-ci est ensuite raffinée par minimisation non-linéaire de la fonction de coût complète (5.15).

L'estimation initiale des paramètres  $\mathbf{a}_{l+1}$ ,  $\mathbf{b}_{l+1}$ ,  $\mathbf{C}_{l+1}$  est obtenue par approximation de l'erreur de reprojection : nous montrons que les directions de déformation  $\mathbf{C}_{l+1}$  peuvent être calculées indépendamment les unes des autres et indépendamment des autres inconnues. Ensuite, une fois les directions de déformation estimées, les coefficients de forme  $\mathbf{a}_{l+1}$  ainsi que les amplitudes de déformation  $\mathbf{b}_{l+1}$  sont évalués.

##### 5.4.4.1 Initialisation des directions

**Séparation des problèmes.** Les directions de déformation  $\{\mathbf{C}_{l+1,j}\}_{j=1}^m$  associées à chaque mode peuvent être estimées de manière indépendante. Pour cela, l'erreur de reprojection est réécrite en utilisant des distances point-droite. La combinaison des équations (5.10) et (5.12), conduit à :

$$\tilde{\mathbf{s}}_{i,j}^{l+1} \sim \underbrace{\tilde{\mathbf{P}}_i \tilde{\mathbf{S}}_{i,j}^l}_{\tilde{\mathbf{s}}_{i,j}^l} + a_{i,l+1} b_{l+1,j} \mathbf{P}_i \mathbf{D}_i \mathbf{C}_{l+1,j}. \quad (5.16)$$

Cette équation représente un point image paramétré par sa position  $a_{i,l+1} b_{l+1,j}$  le long d'une droite, passant par le point  $\tilde{\mathbf{s}}_{i,j}^l$  et avec vecteur directeur  $\mathbf{P}_i \mathbf{D}_i \mathbf{C}_{l+1,j}$ . En remplaçant les points projetés  $\{\tilde{\mathbf{s}}_{i,j}^{l+1}\}_{i,j=1}^{n,m}$  définis par l'équation (5.16) dans chaque terme de l'erreur de reprojection on obtient :

$$\min_{\mathbf{a}_{l+1}, \mathbf{B}_{l+1}} \sum_{i=1}^n \sum_{j=1}^m v_{i,j} d^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{s}}_{i,j}^l + a_{i,l+1} b_{l+1,j} \mathbf{P}_i \mathbf{D}_i \mathbf{C}_{l+1,j}). \quad (5.17)$$

Afin de rendre notre problème indépendant des coefficients de forme  $\{a_{i,l+1}\}_{i=1}^n$  et des amplitudes de déformation  $\{b_{l+1,j}\}_{j=1}^m$ , les distances point-point sont remplacées par des distances point-droite  $d_{pl}^2$ , en introduisant les coordonnées des droites  $\{\tilde{\mathbf{l}}_{i,j}^{l+1}\}_{i,j=1}^{n,m}$  :

$$\tilde{\mathbf{l}}_{i,j}^{l+1} \stackrel{\text{def}}{=} \tilde{\mathbf{s}}_{i,j}^l \times (\mathbf{P}_i \mathbf{D}_i \mathbf{C}_{l+1,j}),$$

ce qui donne :



$$\min_{\mathbf{C}_{l+1}} \sum_{i=1}^n \sum_{j=1}^m v_{i,j} d_{pl}^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{l}}_{i,j}^{l+1}). \quad (5.18)$$

Dans l'équation (5.18), chaque direction  $\mathbf{C}_{l+1,j}$  de  $\mathbf{C}_{l+1}$  est indépendante. Évaluer les directions peut être résolue par  $m$  sous problèmes :

$$\min_{\mathbf{C}_{l+1,j}} \sum_{i=1}^n v_{i,j} d_{pl}^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{l}}_{i,j}^{l+1}). \quad (5.19)$$

**Estimation linéaire.** La première étape permettant d'estimer les directions de déformation consiste à approximer la fonction de coût (5.19) pour aboutir à un problème d'optimisation linéaire au sens des moindres carrés. Pour cela les distances Euclidiennes sont remplacées par des distances algébriques  $d_{A\_pl}^2$  introduites en §1.3) :

$$d_{pl}^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{l}}_{i,j}^{l+1}) \approx \underbrace{\left( \tilde{\mathbf{q}}_{i,j}^T \left[ \tilde{\mathbf{s}}_{i,j}^l \right]_{\times} \mathbf{P}_i \mathbf{D}_i \mathbf{C}_{l+1,j} \right)^2}_{d_{A\_pl}^2(\tilde{\mathbf{q}}_{i,j}, \tilde{\mathbf{l}}_{i,j}^{l+1})}.$$

L'évaluation de  $\mathbf{C}_{l+1,j}$  sous la contrainte  $\|\mathbf{C}_{l+1,j}\| = 1$  est obtenue par SVD de la matrice suivante :

$$\begin{pmatrix} v_{1,j} \tilde{\mathbf{q}}_{1,j}^T \left[ \tilde{\mathbf{s}}_{1,j}^l \right]_{\times} \mathbf{P}_1 \mathbf{D}_1 \\ \vdots \\ v_{n,j} \tilde{\mathbf{q}}_{n,j}^T \left[ \tilde{\mathbf{s}}_{n,j}^l \right]_{\times} \mathbf{P}_n \mathbf{D}_n \end{pmatrix}.$$

Les lignes correspondant à des données manquantes, c'est-à-dire pour lesquelles  $v_{i,j} = 0$ , sont enlevées de la matrice. Notons qu'un point  $j$  doit être visible dans au moins deux vues.

**Raffinement non-linéaire.** La deuxième étape consiste à parfaire l'estimation initiale de chaque  $\mathbf{C}_{l+1,j}$  par optimisation non-linéaire des  $m$  équations (5.19) en utilisant l'algorithme Levenberg-Marquardt décrit en §1.6.2. Cette étape est très peu coûteuse en temps de calcul puisque chaque direction possède 3 paramètres et est estimée indépendamment. Sur ces 3 paramètres, seuls 2 sont indépendants, le troisième étant fixé par la contrainte  $\|\mathbf{C}_{l+1,j}\| = 1$ . Une pénalité  $\left( \|\mathbf{C}_{l+1,j}\|^2 - 1 \right)^2$  est ajoutée aux fonctions de coût (5.19), pour chaque direction de déformation, afin d'imposer cette contrainte.

#### 5.4.4.2 Initialisation des coefficients de forme et des amplitudes de déformation

**Principe.** L'estimation des coefficients de forme  $\mathbf{a}_{l+1}$  et des amplitudes de déformation  $\mathbf{b}_{l+1}$  dépend de tous les paramètres puisque les points image  $\tilde{\mathbf{s}}_{i,j}^{l+1}$  pour chaque vue  $i$  et chaque point  $j$  sont fonctions de  $\mathbf{a}_{l+1} \mathbf{b}_{l+1}^T$ . Nous proposons d'exploiter l'ambiguïté 1D du modèle, décrit en §5.4.2, afin d'estimer linéairement les coefficients de forme et les amplitudes de déformation par normalisation successive de chaque élément contenu dans  $\mathbf{a}_{l+1}$ .

**Les contraintes.** Nous supposons que  $a_{\xi,l+1} \neq 0$  pour  $\xi \in [1, \dots, n]$  et définissons  $\mathbf{a}_{l+1}^\xi \stackrel{\text{def}}{=} \frac{\mathbf{a}_{l+1}}{a_{\xi,l+1}}$  et  $\mathbf{b}_{l+1}^\xi \stackrel{\text{def}}{=} a_{\xi,l+1} \mathbf{b}_{l+1}$ . En ne conservant que les termes de (5.17) dépendant de la vue  $\xi$  on obtient :

$$\min_{\mathbf{b}_{l+1}^\xi} \sum_{j=1}^m v_{\xi,j} d^2(\tilde{\mathbf{q}}_{\xi,j}, \tilde{\mathbf{s}}_{\xi,j}^l + b_{l+1,j}^\xi \mathbf{P}_\xi \mathbf{D}_\xi \mathbf{C}_{l+1,j}). \quad (5.20)$$

Ce problème de minimisation peut être séparé en  $m$  sous problèmes :

$$\min_{\mathbf{b}_{l+1,j}^\xi} v_{\xi,j} d^2(\tilde{\mathbf{q}}_{\xi,j}, \tilde{\mathbf{s}}_{\xi,j}^l + b_{l+1,j}^\xi \mathbf{P}_\xi \mathbf{D}_\xi \mathbf{C}_{l+1,j}), \quad (5.21)$$

revenant chacun à trianguler sur l'image  $\xi$  un point  $j$  appartenant à une droite. L'amplitude de déformation  $b_{l+1,j}^\xi$  est obtenue par *projection orthogonale* de  $\tilde{\mathbf{q}}_{\xi,j}$  sur la droite  $\tilde{\mathbf{l}}_{\xi,j}^{l+1} = \tilde{\mathbf{s}}_{\xi,j}^l \times \mathbf{P}_\xi \mathbf{D}_\xi \mathbf{C}_{l+1,j}$ . Ceci ne peut pas être résolu si  $v_{\xi,j} = 0$ , c'est-à-dire si le point  $j$  n'est pas visible dans la vue  $\xi$ , mais également si la droite  $\tilde{\mathbf{l}}_{\xi,j}^{l+1}$  est mal définie, c'est-à-dire si  $d(\tilde{\mathbf{s}}_{\xi,j}^l, \mathbf{P}_\xi \mathbf{D}_\xi \mathbf{C}_{l+1,j}) < \epsilon_p$ , où  $\epsilon_p$  est un seuil fixé en pratique à quelques pixels. Ce problème apparaît si la direction  $\mathbf{C}_{l+1,j}$  déplace le point  $j$  le long de son rayon de vue associée à la caméra  $\xi$ . A cette étape, nous possédons plusieurs versions de  $\mathbf{b}_{l+1} : \{\mathbf{b}_{l+1}^\xi\}_{\xi=1}^n$  présentant chacune des données manquantes.

**Estimer les  $\mathbf{a}_{l+1}$  et  $\mathbf{b}_{l+1}$ .** De par la relation suivante définissant  $\mathbf{b}_{l+1}^\xi : \mathbf{b}_{l+1}^\xi a_{\eta,l+1} - \mathbf{b}_{l+1}^\eta a_{\xi,l+1} = 0$ , le vecteur  $\mathbf{b}_{l+1}$  concaténant les amplitudes de déformation peut être estimé en évaluant au préalable les coefficients contenus dans  $\mathbf{a}_{l+1}$  :

$$\min_{\mathbf{a}_{l+1}} \sum_{\xi=1}^n \sum_{\eta=1}^n \left\| \mathbf{b}_{l+1}^\xi a_{\eta,l+1} - \mathbf{b}_{l+1}^\eta a_{\xi,l+1} \right\|^2. \quad (5.22)$$

C'est est un problème d'optimisation linéaire sous la contrainte  $\|\mathbf{a}_{l+1}\| = 1$ . Une fois le vecteur des coefficients de forme  $\mathbf{a}_{l+1}$  estimé, les  $\{\mathbf{b}_{l+1}^\xi\}_{\xi=1}^n$  sont denormalisés et moyennés pour aboutir à  $\mathbf{b}_{l+1}$ .

#### 5.4.4.3 Raffinement non-linéaire

Cette étape consiste à minimiser l'équation (5.15) par l'algorithme Levenberg-Marquardt décrit en §1.6.2. La minimisation se fait directement sur les modes de déformation  $\{\mathbf{B}_{l+1,j}\}_{j=1}^m$  ce qui permet d'éviter l'emploi de contraintes pour chaque direction de déformation : ( $\|\mathbf{C}_{l+1,j}\| = 1$ ). Le problème majeur est que  $3m + n$  paramètres doivent être estimés simultanément ce qui peut être coûteux en terme de temps de calcul. Néanmoins, les structures des matrices Jacobienne et Hessienne associées à la fonction de coût sont très creuses, permettant l'emploi de méthodes spécifiques prenant en compte cette propriété pour résoudre les équations normales. Nous utilisons le module de factorisation de Cholesky directement disponible dans le logiciel Matlab<sup>13</sup>. Les temps de calcul s'en trouvent être grandement réduits.

La matrice Jacobienne est constituée de trois parties, comme l'illustre la figure 5.3. La première, relative aux termes de données, est formée de blocs bien agencés. Elle ressemble à celle obtenue lors d'ajustements de faisceaux appliqués sur des scènes rigides. La deuxième partie correspond à la contrainte temporelle. Nous utilisons lors de nos expérimentations un opérateur dérivé du premier ordre aboutissant à une matrice Jacobienne de taille  $((n-1) \times n)$  dont les éléments de la diagonale

<sup>13</sup>Plus précisément, nous utilisons la fonction "mldivide" de Matlab.

principale et ceux de la première diagonale supérieure sont respectivement égaux à -1 et 1. Enfin la troisième partie est relative à la contrainte sur la forme. Sa structure dépend du degré d'interaction entre les points. Elle est très creuse puisque le noyau  $\rho$  utilisé pour construire cette pénalité est local, un point  $n$  n'interagit alors qu'avec ses voisins les plus proches.

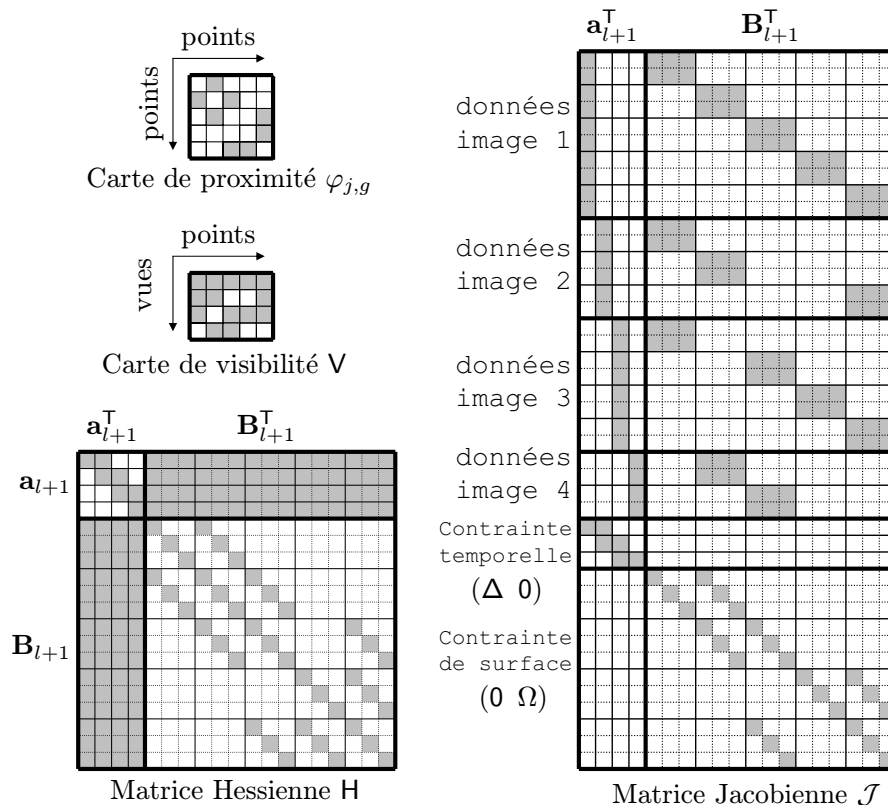


FIG. 5.3 – Structures des matrices Jacobienne et Hessienne sur un exemple synthétique constitué de  $n = 4$  vues et  $m = 5$  points.

### 5.4.5 Critère d'arrêt d'ajout de modes

L'algorithme que nous venons de décrire est basé sur l'ajout itératif de modes au modèle de faible rang. Un critère définissant l'arrêt de ce processus est nécessaire. A chaque ajout d'un mode, le nombre de degré de liberté du modèle augmente et l'erreur de reprojection diminue comme l'illustrent les expérimentations. Les modèles de sélection existant, par exemple BIC ou GRIC, sont mal adaptés à notre problématique. La raison principale est qu'ils sont basés sur une distribution particulière des résidus (Gaussienne). Dans le cadre du modèle de faible rang, les résidus doivent être interprétés différemment; leur dépendance au bruit présent dans les images est plutôt faible. Ils proviennent en grande partie de l'écart entre le modèle de faible rang et la physique qui a engendré l'image, ce qui est difficilement modélisable paramétriquement.

Nous proposons d'utiliser la validation croisée pour sélectionner le nombre de modes. Son principe consiste à partitionner le jeu de données en un jeu d'apprentissage et un jeu de test et d'ensuite moyenner les erreurs obtenues sur les différents jeux de test. Cette approche ne suppose aucune distribution sur les résidus et reflète directement la capacité du modèle à extrapoler à de nouvelles données.

Plus précisément, nous utilisons  $u$  jeux de données obtenus en désactivant certaines composantes de la matrice de visibilité. Les valeurs typiques de  $u$  sont  $u \in [3, \dots, 10]$ , lors de nos expérimentations nous avons choisi  $u = 4$ .

Chaque jeu est un sous-ensemble de la matrice de mesure  $\mathcal{M}$  devant garantir la reconstruction de chaque point dans chaque vue : toutes les lignes et colonnes de  $V$  contiennent au moins deux éléments non nuls. L'erreur associée à chaque jeu de test est obtenue en comparant ses données avec celles issues de la prédiction du modèle évalué sur le jeu d'apprentissage.

Le comportement typique du score de validation croisée est de diminuer jusqu'à ce que le nombre optimal de modes soit obtenu, ensuite, le score augmente. Dans un premier temps, cette descente traduit l'incapacité du modèle à réaliser de bonnes prédictions lorsque le nombre de modes est insuffisant, il est alors trop restrictif. Une fois le nombre optimal de modes atteint, le score augmente car le modèle a alors tendance à expliquer des phénomènes indésirables (bruit sur les données, écart entre le modèle et la physique) : il devient trop flexible pour prédire de nouvelles données.

En pratique ce comportement n'est pas celui observé lorsque les informations *a priori* sont utilisées. Dans ce cas, le score de validation croisée est stable lorsque le nombre de modes est trop important : les contraintes supplémentaires inhibent les degrés de liberté superflus comme l'attestent également les résultats présentés dans (Torresani et al., 2008). Notre critère d'arrêt est donc le suivant, l'ajout de mode est stoppé lorsque le score de validation croisée augmente ou diminue faiblement par rapport à un seuil  $\epsilon$  fixé à  $\epsilon = 10^{-3}$  dans nos expérimentations.

## 5.5 Résultats expérimentaux

Dans cette section, nous allons présenter les différents résultats obtenus sur des données synthétiques et réelles. Les performances de reconstruction de notre algorithme sont comparées avec celles de l'algorithme TORRESANI (Torresani et al., 2008). Ce dernier est décrit en §5.3.2. Pour rappel cet algorithme estime directement la forme explicite du modèle de faible rang. Deux informations *a priori* sont incorporées : les variations temporelles de la surface sont modélisées par un modèle dynamique linéaire appliqué sur les coefficients de forme. Ils sont en outre représentés par une distribution Gaussienne de moyenne nulle. Notons qu'un modèle de projection orthographique est utilisé. Les performances de reconstruction de cet algorithme surpasse celles des approches de reconstruction déjà existantes basées sur le modèle de faible rang, comme l'illustre la partie expérimentale de (Torresani et al., 2008). Le tableau 5.3 résume notre algorithme. Nous utilisons deux variantes : C2F<sup>14</sup> - NO PRIOR qui ne prend pas en compte les informations *a priori* décrites en §5.4.2 et C2F - PRIORS qui les utilisent.

### 5.5.1 Données synthétiques

Nous avons généré un jeu de données synthétiques en utilisant le modèle de visage *Candide-3* décrit en §2.4.2. Un modèle de caméra perspectif est utilisé pour projeter les formes 3D constituant la vérité terrain. La matrice de mesure est composée de  $n = 70$  images et  $m = 113$  points ; un bruit Gaussien de variance 2 pixels est appliqué aux points 2D. Afin de comparer les performances des différents algorithmes, l'erreur de reprojection, le score de validation croisée ainsi que l'erreur 3D sont mesurés en fonction du nombre de modes et du taux de données manquantes. Notons que

<sup>14</sup>"Coarse to Fine" en anglais.

**OBJECTIF**

Soit un ensemble de points image mis en correspondance  $\{\tilde{\mathbf{q}}_{i,j}\}_{i,j=1}^{n,m}$  et un ensemble de caméras  $\{\tilde{\mathbf{P}}_i\}_{i=1}^n$  calibrées de manière quelconque. L'objectif est de calculer, pour chaque image  $i$ , le déplacement global  $\tilde{\mathbf{D}}_i$  et les déformations  $\{\tilde{\mathbf{S}}_{i,j}^l\}_{j=1}^m$  de la surface 3D observée. Ces dernières sont représentées par un modèle de faible rang dont les modes sont estimés hiérarchiquement : l'erreur de reprojection associée aux  $\{\tilde{\mathbf{S}}_{i,j}^{l+1}\}_{i,j=1}^{n,m}$  est plus petite que celle introduite par les  $\{\tilde{\mathbf{S}}_{i,j}^l\}_{i,j=1}^{n,m}$ . Le nombre de modes est sélectionné par validation croisée.

**ALGORITHME**

**Estimation de la forme moyenne  $\{\tilde{\mathbf{M}}_j\}_{j=1}^m$  et des déplacements globaux  $\{\tilde{\mathbf{D}}_i\}_{i=1}^n$**

1. (§5.4.3) Lancer une reconstruction rigide de la scène. Les points image  $\{\tilde{\mathbf{q}}_{i,j}\}_{i,j=1}^{n,m}$ , la carte de visibilité  $V$  et les paramètres internes des caméras  $\{\mathbf{K}_i\}_{i=1}^n$  sont les données d'entrée. La forme moyenne  $\{\tilde{\mathbf{M}}_j\}_{j=1}^m$  ainsi que les déplacements globaux  $\{\tilde{\mathbf{D}}_i\}_{i=1}^n$  sont obtenus en sortie.
2. (§5.4.5) Calculer le score  $\chi_0$  de validation croisée et initialiser le nombre de modes  $l \leftarrow 0$ .
3. Initialiser la forme 3D pour chaque image  $i$  avec la forme moyenne :  $\tilde{\mathbf{S}}_{i,j}^l \leftarrow \tilde{\mathbf{M}}_j$ .

**Triangulation itérative des modes de déformation**

1. (§5.4.4.1) Initialiser les directions de déformation  $\mathbf{C}_{l+1}$ .
2. (§5.4.4.2) Calculer les coefficients de forme  $\mathbf{a}_{l+1}$  et les amplitudes de déformation  $\mathbf{b}_{l+1}$ .
3. (§5.4.4.3) Raffinement non-linéaire de  $\mathbf{a}_{l+1}$ ,  $\mathbf{b}_{l+1}$ ,  $\mathbf{C}_{l+1}$  par minimisation de l'erreur de reprojection.
4. (§5.4.5) Calculer le score de validation croisée  $\chi^{l+1}$ .
5. (§5.4.5) Arrêt de l'algorithme si  $\chi_l - \chi_{l+1} \leq \varepsilon$ .
6. Mettre à jour la structure 3D :  $\tilde{\mathbf{S}}_{i,j}^{l+1} \leftarrow \tilde{\mathbf{S}}_{i,j}^l + a_{i,l+1} b_{l+1,j} \tilde{\mathbf{C}}_{l+1,j}$ .
7. Incrémenter le nombre de modes  $l \leftarrow l + 1$  et reprendre à l'étape 1 du processus de triangulation.

TAB. 5.3 – Les différentes étapes de l'estimation hiérarchique du modèle de faible rang. Les informations *a priori*, décrites en §5.4.2, sont éventuellement prises en compte à l'étape 3 du processus de triangulation.

l'erreur 3D est mesurée après avoir compensé une transformation de similarité entre la vérité terrain et les formes reconstruites.

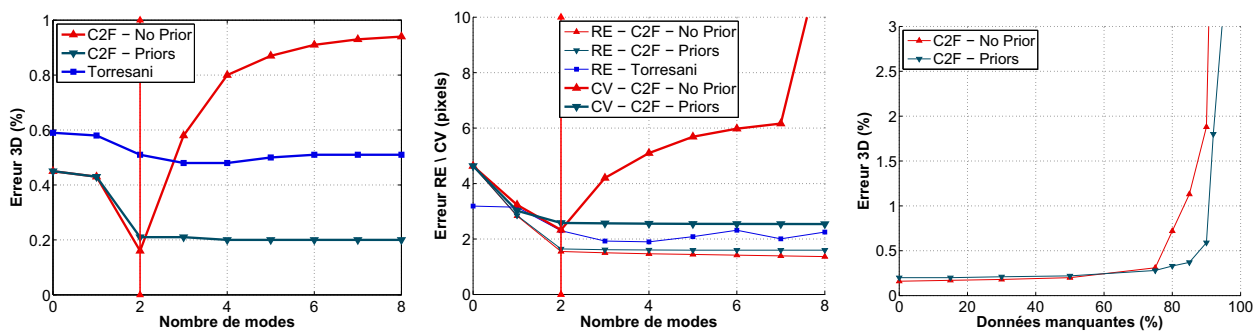


FIG. 5.4 – Comparaison sur des données de synthèse des algorithmes TORRESANI, C2F - NO PRIOR et C2F - PRIORS. A gauche : l'erreur 3D en fonction du nombre de modes. Au centre : l'erreur de reprojection (RE) et le score de validation croisée (CV) en fonction du nombre de modes. A droite : l'erreur 3D en fonction du pourcentage de données manquantes.

La première expérience, illustrée sur la figure 5.4 (à gauche), traduit l'influence du nombre de modes sur l'erreur 3D. Les performances de C2F - NO PRIOR se dégradent lorsqu'un nombre de modes trop important est utilisé : au delà de 2 modes, l'erreur 3D augmente rapidement. C2F - PRIORS et TORRESANI sont beaucoup moins sensibles à ce phénomène, les erreurs 3D associées stagnent pour un nombre trop important de modes. Les déformations sont estimées plus précisément avec les approches proposées qu'avec TORRESANI lorsque le nombre de mode est optimal : l'erreur 3D est respectivement de 0,17%, 0,2%, 0,48% pour C2F - NO PRIOR, C2F - PRIORS et TORRESANI. C2F - PRIORS offre globalement les meilleures performances de reconstruction en fonction du nombre de modes.

La figure 5.4 (au centre) montre l'évolution de l'erreur de reprojection et du score de validation croisée en fonction du nombre de modes. Comme pressentie, l'erreur de reprojection décroît lorsque le nombre de degrés de liberté augmente tandis que le score de validation croisée se comporte sensiblement comme l'erreur 3D. Ce dernier permet de sélectionner le nombre optimal de modes pour C2F - NO PRIOR alors que pour C2F - PRIORS ce nombre est légèrement sous-estimé. Il n'y a pas cependant de dégradation significative de la forme 3D reconstruite : l'erreur 3D optimale est de 0,2% et celle associée au mode sélectionné est de 0,21%. Le taux de succès de la validation croisée pour l'algorithme C2F - NO PRIOR, c'est-à-dire sa capacité à sélectionner le nombre optimal de modes, est de 94%, 89% et 88% pour respectivement aucune donnée manquante, 25% et 50% de données manquantes. Ces taux sont obtenus sur 100 tirages des jeux de test et d'apprentissage. Ils prouvent la pertinence de la validation croisée pour la sélection du nombre de modes. Ceci est d'autant plus vrai que le nombre de modes est sous ou sur estimé de 1 en général.

Enfin, nous avons testé la robustesse des différents algorithmes aux données manquantes. Pour cela un certain pourcentage de données n'est pas pris en compte pour trianguler les modes. En pratique, on fixe des éléments de  $V$  à 0 sous contraintes qu'un point 3D de la surface soit visible dans au moins 3 vues et qu'au moins 3 points 3D soit visibles par chaque caméra. Les données manquantes sont tirées aléatoirement sans consistance temporelle et spatiale pouvant traduire des pertes ponctuelles du suivi mais ne reflètent pas la présence d'occultations. La robustesse de notre algorithme aux auto-occultations et occultations externes est démontrée sur des données réelles en §5.5.2. L'al-

gorithme C2F - PRIORS estime correctement la forme 3D jusqu'à 92% de données manquantes tandis que C2F - NO PRIOR gère jusqu'à 90%, comme l'illustre la figure 5.4. Ces performances s'expliquent en grande partie par le bon comportement de la validation croisée qui sélectionne la plupart du temps le nombre optimal de modes. L'algorithme TORRESANI diverge dans la majorité des cas ; il n'est pas représenté sur le graphe.

**Discussion.** Nous avons également testé l'algorithme TORRESANI sur des données issues de la projection du modèle *Candide-3* par un modèle de caméra orthographique. Les résultats obtenus ne sont pas représentés sur les graphes de la figure 5.4 afin de ne pas prêter à confusion. En effet, ils ne sont pas directement comparables avec ceux de la figure 5.4 : les données d'entrées fournies à l'algorithme sont différentes. Les performances de TORRESANI sur ces données sont très bonnes, la surface 3D reconstruite est très proche de la vérité terrain : l'erreur 3D est de 0,09%. La robustesse aux données manquantes est également bien meilleure : la reconstruction est correcte jusqu'à environ 80%. Néanmoins, nous avons pu observer au cours des différentes expérimentations, que les performances de cette algorithme se dégradent fortement en présence de l'effet perspectif ce qui est plutôt gênant pour estimer les déformations à partir d'une vidéo réelle. Dans (Torresani et al., 2008), les résultats présentés sont issus uniquement de données provenant de capteur de mouvement qui sont projetées artificiellement par une caméra orthographique.

## 5.5.2 Données réelles

**La vidéo du visage.** Cette vidéo est obtenue par projection perspective des données 3D utilisées dans (Torresani et al., 2008)<sup>15</sup>. Ces dernières proviennent d'un système de capture de mouvement observant un visage sur lequel 40 marqueurs sont placés. La vidéo est composée de 316 images, les déformations sont plutôt faibles et concentrées principalement sur la région de la bouche comme l'illustre la figure 5.5. De bons résultats sont obtenus sur ces données par l'algorithme TORRESANI dans le cadre d'une projection orthographique, comme rapporté dans (Torresani et al., 2008). Nous souhaitons démontrer que notre approche permet de reconstruire une surface déformable filmée par une caméra perspective, contrairement à la majorité des approches existantes, dont TORRESANI. Pour cela, un modèle de caméra perspectif avec une focale de 660 millimètre est utilisée. L'erreur 3D obtenue par C2F - NO PRIOR décroît quelque soit le nombre de modes utilisés, tandis que l'algorithme TORRESANI ne capture quasiment aucune déformation, comme le représente la figure 5.6(b). En effet, l'emploi d'un modèle de caméra orthographique implique que la variance de l'erreur présente dans les images provient majoritairement de l'effet perspectif, difficilement interprétable linéairement, et non pas des déformations du visage. La présence d'informations *a priori* n'est pas requise en raison des faibles déformations du visage et de la quasi absence de bruit dans les données. La figure 5.6(a) représente le premier mode du modèle de faible rang estimé par notre approche. Les déformations capturées se situent principalement au niveau de la bouche, comme pressenti.

**La vidéo de la feuille de papier.** Cette vidéo est composée de 203 images de taille  $720 \times 576$ . L'algorithme de recalage d'images décrit au chapitre 4 est utilisé pour suivre les déformations de la surface ; 140 correspondances de points sont ainsi engendrées. La figure 5.7 illustre les résultats obtenus. Les deux algorithmes proposés C2F - NO PRIOR et C2F - PRIORS, sélectionnent respectivement

<sup>15</sup>Nous remercions L. Torresani pour avoir mis à disposition son code. Ce dernier est disponible à l'adresse <http://research.microsoft.com/ltorre/nrsfm.aspx>.

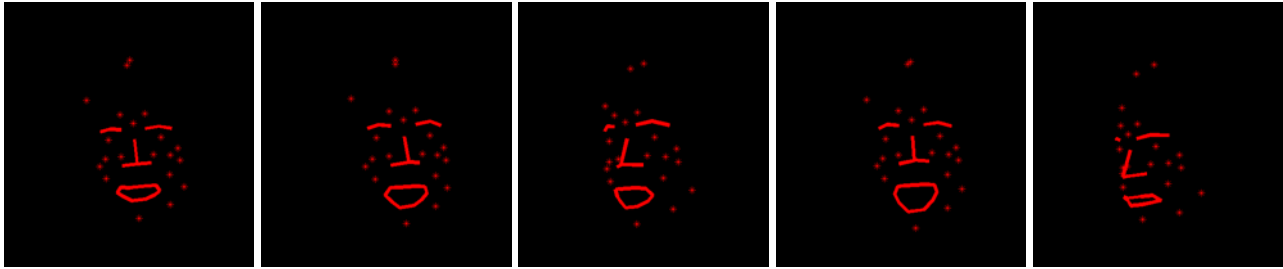


FIG. 5.5 – Images extraites de la vidéo du visage.

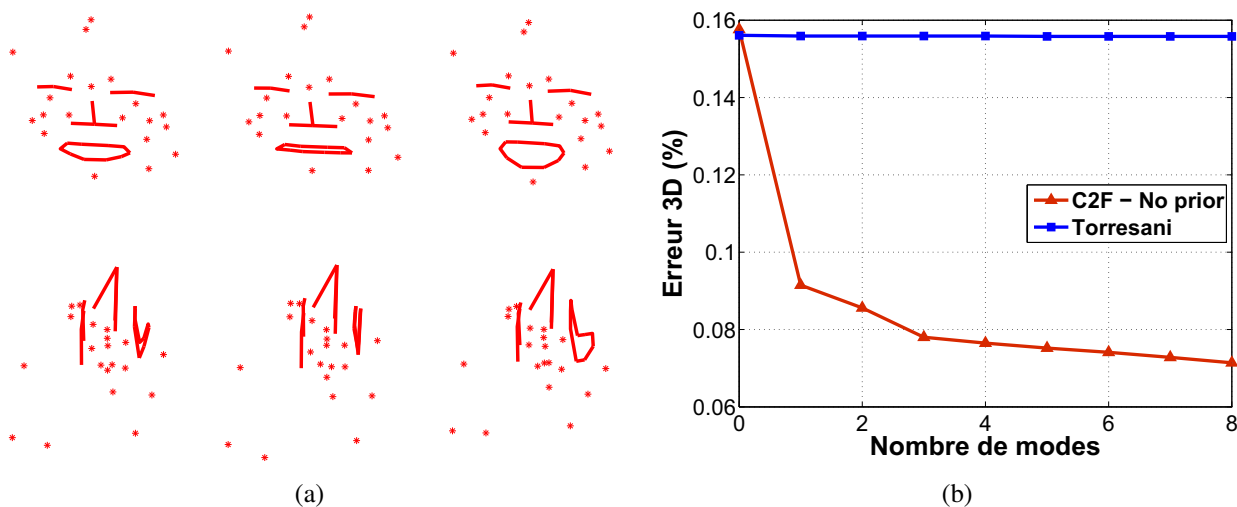


FIG. 5.6 – **Résultat de la reconstruction de la vidéo du visage.** (a) A gauche : la forme moyenne. Au centre : la forme moyenne à laquelle est ajouté le premier mode estimé par notre approche. A droite : la forme moyenne à laquelle est soustrait le premier mode estimé par notre approche. (b) Évolution de l'erreur 3D en fonction du nombre de modes pour les algorithmes C2F - NO PRIOR et TORRESANI.

0 et 3 mode(s), l'erreur de reprojection est respectivement de 5,10 et 0,84 pixels. C2F - NO PRIOR fonctionne très mal sur cette vidéo. Les déformations estimées sont biaisées : elles ne correspondent pas à la réalité. Ceci démontre que l'emploi d'informations *a priori* ne peut pas être évité. En présence de contraintes supplémentaires, la reconstruction est visuellement correcte. En outre, le score de validation croisée est de 1,82 pixels ce qui garantit une bonne prédiction des nouvelles données. Notons que l'algorithme TORRESANI a également été testé sur cette vidéo. Les déformations ne sont que très faiblement capturées. Certains modes encodent un déplacement rigide de la surface. L'erreur résiduelle est de 2,1 pixels pour 3 modes.

Une occultation externe est simulée sur cette vidéo, en ne prenant pas en compte 24 points adjacents sur 120 images consécutives, soit environ 10% du jeu de données. C2F - PRIORS sélectionne 3 modes. L'erreur de reprojection est de 1,44 pixels et le score de validation croisée de 1,82 pixels, soit très légèrement supérieur au score obtenu lorsque toutes les données sont prises en compte. Les déformations estimées sont quant à elles semblables à celles évaluées précédemment, comme l'illustre la figure 5.8.



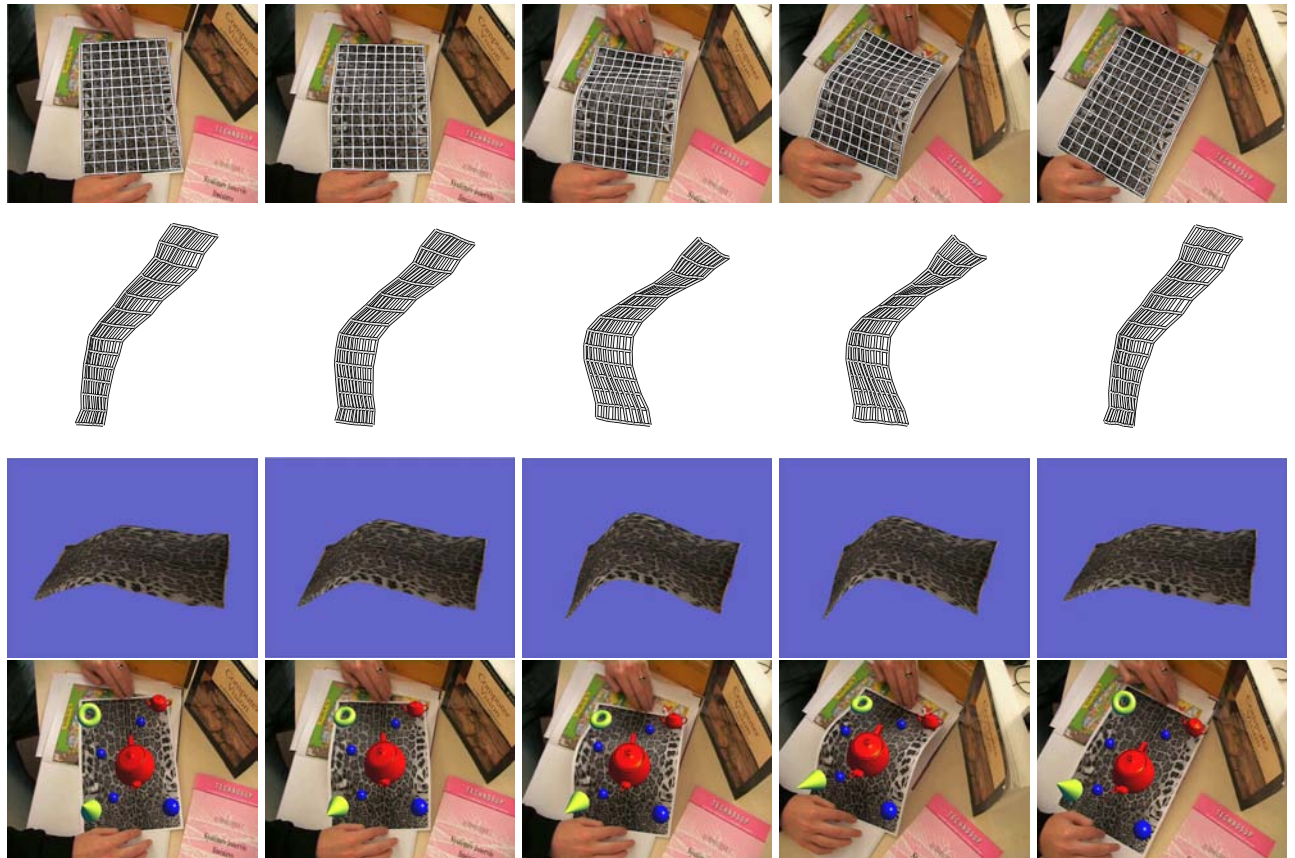


FIG. 5.7 – **Résultat de la reconstruction sur la vidéo de la feuille de papier.** 1<sup>ère</sup> ligne : images extraites de la vidéo ainsi que les points suivis. 2<sup>ème</sup> ligne : résultat de la reconstruction par C2F - PRIORS. 3<sup>ème</sup> ligne : création synthétique de nouvelles vues (réalisée par Mathieu Perriollat). 4<sup>ème</sup> ligne : augmentation 3D de la vidéo (réalisée par Umberto Castellani).

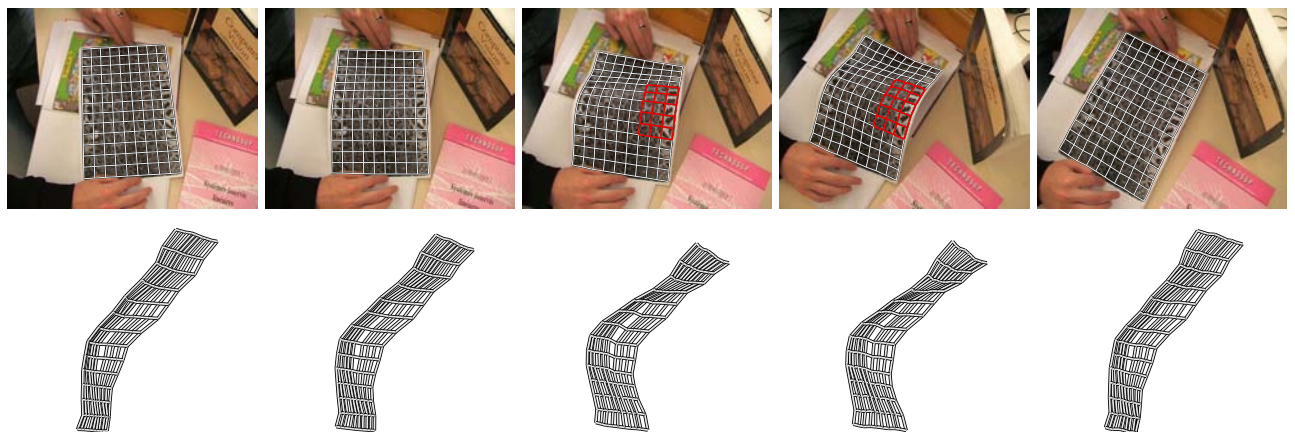


FIG. 5.8 – **Résultat de la reconstruction sur la vidéo de la feuille de papier en présence de données manquantes.** En haut : images extraites de la vidéo. Les correspondances utilisées et celles non utilisées sont respectivement représentées en blanc et en rouge (gris). En bas : résultat de la reconstruction par C2F - PRIORS.

**La vidéo "Desperate Housewives".** Nous avons extrait 100 images de taille  $624 \times 352$  de la série "Desperate Housewives" représentant le personnage de Gabrielle Solis. La vidéo est illustrée sur la figure 5.9.



FIG. 5.9 – Images extraites de la vidéo "Desperate Housewives".

Les déformations du visage dans les images sont suivies à l'aide d'un modèle actif d'apparence (AAM<sup>16</sup>), présenté en §2.4. Les 68 sommets constituant l'AAM sont reconstruits avec notre algorithme. La figure 5.10 présente les résultats obtenus par C2F - PRIORS, C2F - NO PRIOR et C2F - PRIORS trouvent que 3 modes sont nécessaires pour capturer les déformations. Ils obtiennent respectivement 0,82 et 0,97 pixels comme erreur de reprojection et, 1,21 et 1,22 pixels pour le score de validation croisée comme rapporté sur la figure 5.11. Ces valeurs prouvent que le modèle reconstruit est capable de prédire correctement de nouvelles données. Dans cet exemple, l'information *a priori* qu'un visage est présent dans la vidéo est utilisée uniquement lors du suivi : notre algorithme reconstruit les déformations du visage de manière générique.

**La vidéo de la feuille de papier auto-occultée.** Cette vidéo est constituée de 338 images de taille  $720 \times 576$ . Une partie de la surface est auto-occultée entre les images 89 et 266. L'algorithme de recalage d'images d'une surface déformable auto-occultée, proposé au chapitre 4, est utilisé pour suivre les déformations de la surface. Ce dernier délivre 60 correspondances de points ainsi qu'une carte d'auto-occultation probabiliste à partir de laquelle est extraite la carte de visibilité  $V$  relative aux correspondances. Les résultats du recalage et de la détection des auto-occultations sont illustrés sur la figure 5.12.

Ces informations sont utilisées comme données d'entrée de l'algorithme C2F - PRIORS. Afin d'améliorer la qualité de la reconstruction, nous avons incorporé des informations *a priori* supplémentaires au traitement de cette vidéo. La forme 3D est supposée connue sur la première et la dernière image de la vidéo : elle correspond à une surface plane. Cette hypothèse est également utilisée dans (Salzmann et al., 2007a,b). Les résultats de la reconstruction par C2F - PRIORS sont représentés sur la figure 5.13. La forme 3D apparaît être cohérente avec les déformations présentes dans les images. Pour la première et la dernière images de la vidéo, une surface plane est reconstruite lorsque les informations *a priori* décrites ci-dessus sont incorporées au processus de reconstruction, dans le cas contraire, les formes reconstruites sont légèrement bombées. Enfin, les données manquantes sont correctement prédites en raison des contraintes imposées sur les variations temporelles et spatiales de la surface engendrée par le modèle de faible rang.

<sup>16</sup>"Active Appearance Model".

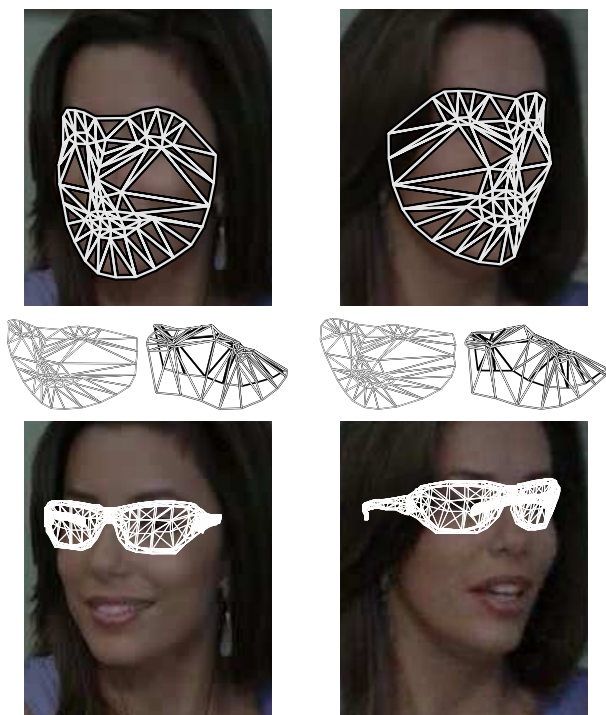


FIG. 5.10 – Résultat de la reconstruction sur la vidéo "Desperate Housewives". En haut : suivi des déformations 2D par un modèle AAM (réalisé par Julien Peyras). Au centre : reconstruction 3D des sommets du modèle. En bas : augmentation 3D des images (réalisée par Julien Peyras).

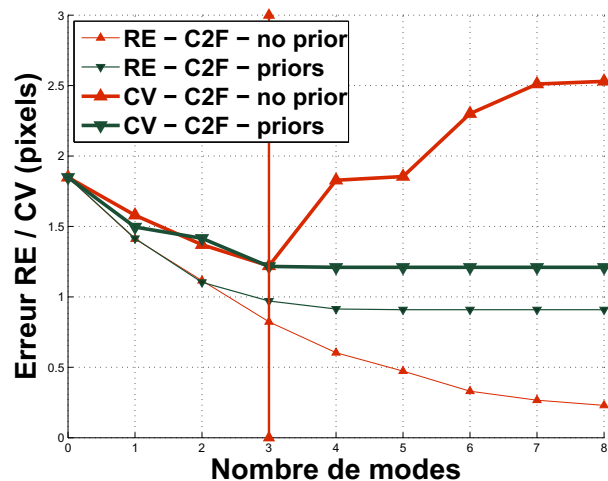


FIG. 5.11 – Erreur de reprojection (RE) et score de validation croisée (CV) sur la vidéo "Desperate Housewives" en fonction du nombre de modes. Les algorithmes C2F - NO PRIOR et C2F - PRIOR sont testés. Ils trouvent que 3 modes sont requis pour capturer les déformations.

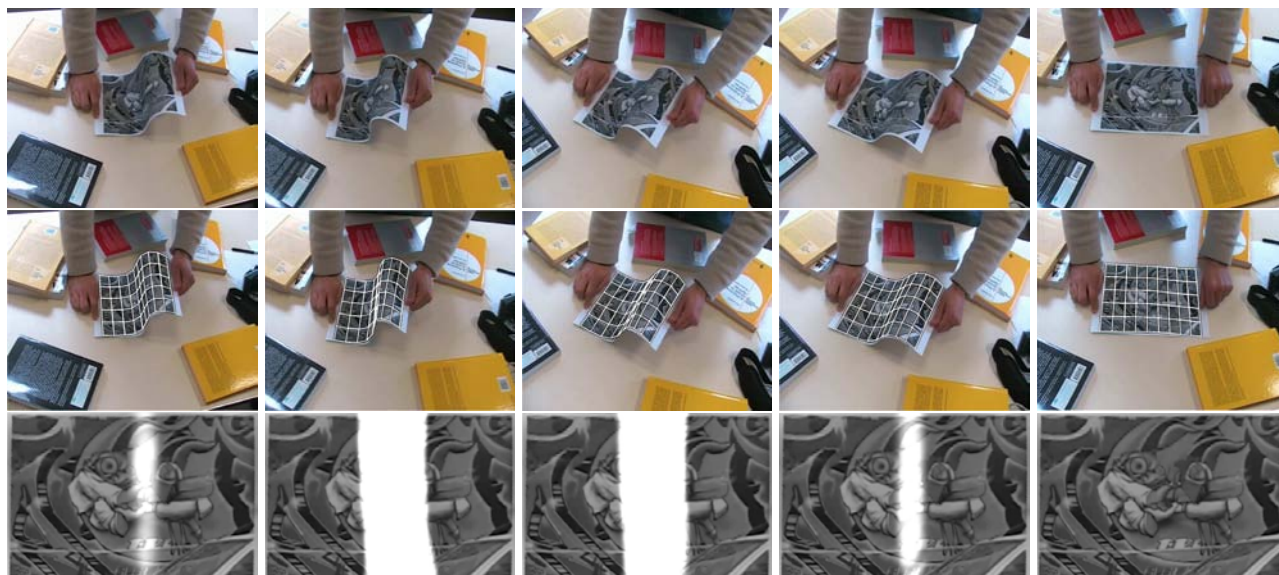


FIG. 5.12 – La vidéo de la feuille de papier auto-occultée et les données associées. En haut : images extraites de la vidéo. Au milieu : résultat du recalage par l'algorithme présenté au chapitre 4. Les correspondances de points sont obtenues après sous échantillonnage du champ de déplacement dense. En bas : la carte d'auto-occultation probabiliste estimée pour chaque image.

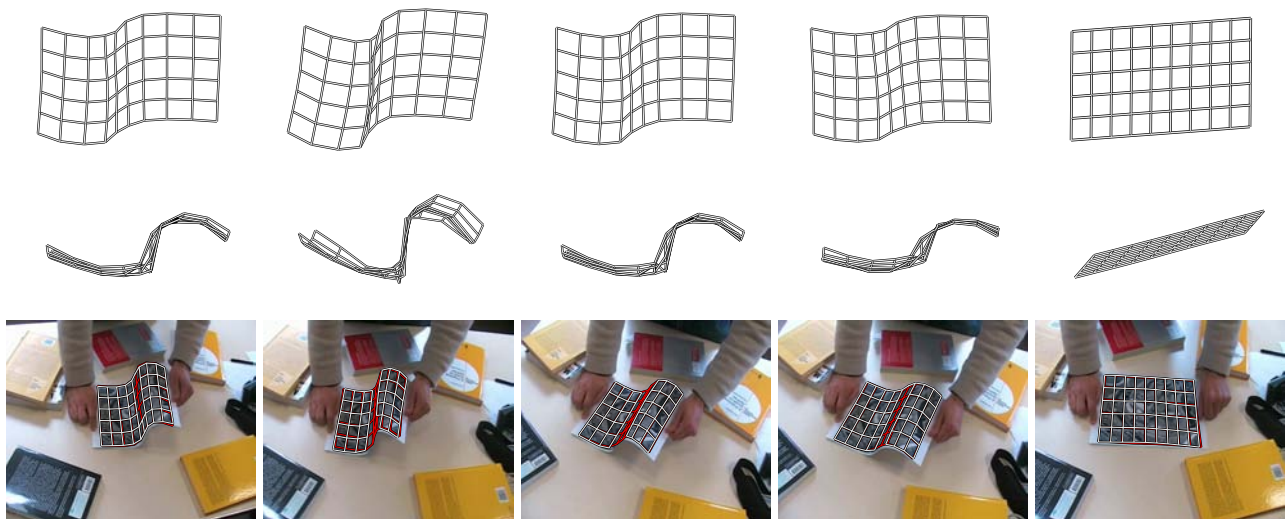


FIG. 5.13 – **Résultat de la reconstruction par C2F - PRIORS sur la vidéo de la feuille de papier auto-occultée.** En haut : les formes 3D reconstruites visualisées de face. Au milieu : les formes 3D reconstruites visualisées de profil. En bas : en rouge la reprojection des formes 3D reconstruites et en blanc les maillages 2D de référence.

## 5.6 Conclusion

Nous proposons dans ce chapitre un algorithme de reconstruction 3D basé sur le modèle de faible rang. Ce dernier est vu comme un ensemble de modes de déformation ordonnés, estimés hiérarchiquement. Il en résulte un algorithme qui gère les données manquantes, utilise un modèle de caméra perspectif et sélectionne automatiquement le nombre optimal de modes de déformation par validation croisée. Les résultats expérimentaux sur des données synthétiques montrent que le nombre de modes sélectionné correspond au minimum de l'erreur 3D. Nous avons également incorporé deux informations *a priori* supplémentaires dans la fonction de coût. Ces dernières améliorent grandement la qualité de la reconstruction. La principale observation que nous faisons est que la validation croisée semble être un moyen performant pour sélectionner le nombre de modes puisque l'évolution de l'erreur 3D ressemble à celui du score de validation croisée. Nous envisageons dans les travaux futurs de sélectionner automatiquement les poids associés aux informations *a priori*. Dans la littérature, ils sont fixés soit de manière heuristique soit par essai et erreur comme ce fut le cas lors de nos expérimentations. Une solution possible est de minimiser le score de validation croisée sur ces poids. Il n'est pas clair que ceci puisse être réalisé dans des temps de calcul raisonnables.

# Conclusion

**Travaux effectués.** Les travaux que nous venons de présenter traitent du recalage d'images d'une surface déformable et de la reconstruction 3D de surfaces déformables. Trois contributions majeures ont été introduites au cours du tapuscrit. Tout d'abord, des algorithmes permettant de recaler efficacement des images d'une surface déformable ont été proposés. Puis nous nous sommes intéressés à la gestion des auto-occultations, ces dernières ayant jusqu'alors peu été prises en compte dans la littérature. Ces différentes techniques fournissent en sortie un champ de déplacement dense utilisé, après sous-échantillonnage, comme entrée d'un algorithme de reconstruction 3D basé sur le modèle de faible rang. Nous avons proposé une nouvelle représentation de ce modèle en ordonnant les modes de déformation. La qualité de la reconstruction s'en trouve être grandement améliorée.

Nous avons principalement appliqué nos travaux à la réalité augmentée : augmentation 2D d'images d'une surface auto-occultée et augmentation 3D de surfaces déformables sur une vidéo. Les algorithmes que nous avons introduits se limitent pour l'instant aux traitements de vidéos en post production en raison des paramètres fixés empiriquement et des temps de calcul requis. Nous donnons quelques pistes au paragraphe « perspectives » ci-dessous afin d'étendre nos travaux sur la gestion des auto-occultations à des applications traitant un flux vidéo en temps réel.

- ▷ **Recalage efficace d'images d'une surface déformable.** Nous avons proposé plusieurs algorithmes permettant un recalage efficace d'images d'une surface déformable. Ils sont basés sur la paramétrisation par primitives des déformations, permettant l'emploi d'algorithmes compositionnels en relaxant leur contrainte de groupe. Ainsi, les algorithmes de composition inverse et ceux basés apprentissage ont été étendus aux fonctions de type plaque mince. Nous avons également proposé une amélioration des approches basées apprentissage, en apprenant une relation linéaire par morceaux. La combinaison du concept de déformations guidées par primitives avec l'approche basée apprentissage proposée permet de surpasser les algorithmes existants sur la plupart des expérimentations.
- ▷ **Gestion des auto-occultations en recalage d'images.** Nous avons proposé une approche originale pour le recalage d'images d'une surface déformable auto-occultée. Le cadre proposé s'appuie sur deux composantes principales. Tout d'abord, la fonction de déformation est contrainte à se comprimer le long de la frontière d'auto-occultation, puis cette propriété de contraction est utilisée pour définir un module de détection de pixels auto-occultés. Il en résulte une modélisation explicite des auto-occultations. Nous avons présenté de nombreux résultats sur des vidéos complexes attestant de la validité de notre approche. Les algorithmes existants, ne modélisant pas explicitement les auto-occultations, échouent sur ces données.
- ▷ **Reconstruction 3D de surfaces déformables.** Nous avons proposé un algorithme de re-

construction 3D basé sur le modèle de faible rang. Nous représentons ce dernier comme un ensemble de modes ordonnés, estimés hiérarchiquement. Il en résulte un algorithme qui gère les données manquantes, utilise un modèle de caméra perspectif et sélectionne automatiquement le nombre optimal de modes de déformation par validation croisée. Toutes ces propriétés n'avaient pas été regroupées dans un seul algorithme auparavant. Nous avons testé et comparé l'approche proposée sur des vidéos synthétiques et réelles : les déformations reconstruites sont visuellement proches de la réalité, permettant l'augmentation 3D réaliste de surfaces déformables sur une vidéo.

**Perspectives.** Quelques améliorations doivent encore être apportées avant d'aboutir à un système de réalité augmentée transférable dans le domaine industriel. L'algorithme permettant l'augmentation 3D d'images d'une surface déformable se limite par nature aux traitements de vidéos en post production. Néanmoins, il est envisageable d'apprendre les modes de déformation sur les premières images de la vidéo. Le modèle appris (morphable) peut ensuite être estimé à partir d'une seule image (Salzmann et al., 2005). Le flux vidéo sera alors traité au fur et à mesure que les images arrivent.

L'algorithme de recalage gérant les auto-occultations doit être perfectionné pour pouvoir traiter un flux vidéo temps réel. Pour cela, les paramètres inhérents aux différentes composantes de la fonction de coût devront être sélectionnés automatiquement, les temps de calcul réduits et les changements d'éclairage pris en compte.

- ▷ **Robustesse aux changements d'éclairage.** Incorporer explicitement un modèle d'illumination local dans le terme de données introduit des ambiguïtés, les variations locales d'intensité pouvant alors être expliquées par le modèle d'illumination et par la fonction de déformation. Une solution possible consisterait à recalibrer les images dans un espace invariant à ces phénomènes, comme proposé dans (Pizarro and Bartoli, 2007), ce qui nécessite l'estimation de paramètres supplémentaires de calibration photométrique. Cette approche a récemment été étendue aux modèles actifs d'apparence (Pizarro et al., 2008).
- ▷ **Réduction des temps de calcul.** Il paraît indispensable d'accélérer le processus de recalage d'images d'une surface déformable auto-occultée. Les déformations extrêmes introduites par les auto-occultations font que l'approximation de la composition des algorithmes guidés par primitives proposé au chapitre 3 n'est plus valide. Ces algorithmes ne peuvent donc pas être utilisés. Un gain substantiel serait obtenu en réduisant le nombre d'informations image utilisées. En effet, la prise en compte de tous les pixels engendre des matrices Jacobienne et Hessienne dont le nombre d'entrées est extrêmement important, rendant le processus d'optimisation très lourd. Un critère basé primitives apparaît être exclu : le nombre de primitives mises en correspondance sera probablement insuffisant, notamment au voisinage de la frontière d'auto-occultation, pour définir correctement la fonction de déformation. Une alternative pourrait consister à utiliser des « patches » d'intérêt. Ils constituent un bon compromis entre précision et efficacité. Leur répartition sur l'image de texture ainsi que leur nombre sont des paramètres critiques dont l'influence devra être étudiée.
- ▷ **Sélection automatique des paramètres.** Cette tâche apparaît indispensable pour aboutir à un système de réalité augmentée entièrement automatique. Elle reste un problème ouvert même si

de nombreux travaux l'ont abordé en proposant différents critères comme par exemple la validation croisée (Bartoli, 2007), les "L-curves" (Hansen, 2005) et la "L-tangent norm" (Brunet et al., 2008) ont été proposés dans la littérature. Ils permettent d'estimer le poids  $\lambda_s$  contrôlant le lissage spatial d'une fonction de coût constituée de deux termes, et requièrent en outre des correspondances exactes. Leur extension à des fonctions de coût constituées de plusieurs termes et à un critère direct nécessite des études complémentaires.





# Annexe A

## Capture de déformations à partir de données issues d'un capteur 3D

---

---

Cette annexe aborde le problème de la capture des déformations d'une surface à partir de données issues d'un capteur 3D, comme par exemple un scanner à *lumière structurée* ou une paire d'images *stéréo*. L'emploi de données 3D permet de limiter l'influence de la texture de la surface sur la qualité des déformations capturées. Ainsi, les surfaces peu texturées peuvent également être traitées. L'algorithme proposé déforme un maillage régulier jusqu'à ce qu'il corresponde « au mieux » au nuage de points 3D. Pour cela une fonction de coût constituée de plusieurs termes de données et de régularisation est minimisée. L'estimation des déformations peut être faite très efficacement en combinant des méthodes d'optimisation de moindres carrés non-linéaires avec des opérateurs spécifiques aux matrices creuses. L'algorithme proposé est également rendu robuste afin de gérer la présence de données manquantes et erronées. Des résultats expérimentaux sur des données bruitées provenant de différents capteurs sont présentés. L'estimation des déformations s'avère être robuste aux données aberrantes et raisonnablement efficace en terme de temps de calcul. Ces travaux ont été publiés dans (Castellani et al., 2007) et (Castellani et al., To appear).

---

---

### A.1 Introduction

L'objectif de ces travaux est d'aboutir à un système permettant la capture des déformations d'une surface telle une page de livre que l'on tournerait en face d'un capteur 3D. Les applications possibles sont la compression de données 3D, la réalité augmentée, l'animation de surfaces déformables et la synthèse d'images. Nous verrons par exemple que des vidéos peuvent être générées synthétiquement à partir des déformations capturées, la texture appliquée sur la surface étant laissée au choix de l'utilisateur.

Tous ces domaines d'application nécessitent un recalage précis des données 3D au cours du temps ainsi que la reconstruction<sup>1</sup> de la surface. Les avantages d'utiliser des données 3D sont multiples :

---

<sup>1</sup>Dans cette annexe, le terme « reconstruction » n'as pas le même sens qu'au chapitre 5. La surface « reconstruite » traduisant la surface « interpolant » le nuage de points 3D.

- ▷ **Évaluation 3D de la surface.** Des déformations 3D réelles sont obtenues, par opposition aux champs de déplacement 2D issus d'algorithmes de recalage d'images comme par exemple ceux proposés aux chapitres 3 et 4.
- ▷ **Sensibilité à la texture.** Contrairement à la plupart des autres méthodes, nous n'utilisons pas de points d'intérêt mis en correspondance (Guskov et al., 2003; Scholz et al., 2005) ou bien encore de flot optique (Scholz and Magnor, 2004; Yamamoto et al., 1990) : la qualité de la reconstruction est alors indépendante de la texture présente sur la surface. En particulier, les surfaces peu texturées font échouer les méthodes existantes, notamment celle présentée au chapitre 5, tandis que l'algorithme proposé les gère naturellement. Notons que récemment, une approche permettant la reconstruction 3D de surfaces peu texturées à partir de données image est proposée dans (Salzmann et al., 2008). Leur algorithme est basé sur l'apprentissage de modèles locaux.

Les données fournies à l'entrée de l'algorithme proposé sont, des séquences de nuages de points 3D ainsi que les contours approximatifs de la région d'intérêt. L'algorithme proposé déforme un maillage régulier de manière robuste jusqu'à ce qu'il corresponde « parfaitement » au nuage de points 3D et ceci en dépit de la présence inévitable de données aberrantes et manquantes. Les contributions sont doubles :

Tout d'abord, nous montrons que notre problématique est correctement modélisée en déformant sous différentes contraintes un maillage régulier. La fonction de coût introduite est composée de plusieurs termes de régularisation et de données. Ces derniers s'appuient sur la notion de distance entre le modèle et le nuage de points 3D. Les correspondances sont déterminées par la notion de point le plus proche introduite par l'opérateur min. Cet opérateur est explicitement embarqué dans les termes de données ce qui permet d'éviter un processus d'optimisation en deux étapes inhérent aux algorithmes de type "Itérative Closest Point" (ICP). Un estimateur robuste est de plus inclus dans les termes de données garantissant la robustesse de notre algorithme aux données manquantes et erronées. Les termes de régularisation pénalisent les variations spatiales et temporelles de la surface. Ils limitent les ambiguïtés de reconstruction en contraignant l'espace des déformations admissibles.

L'algorithme Levenberg-Marquardt est utilisé pour minimiser la fonction de coût combinée en prenant en compte la nature extrêmement creuse des matrices Jacobiennes associées à chaque terme. L'estimation des déformations est alors efficace en terme de temps de calcul. L'emploi de la transformée en distance (Fitzgibbon, 2003) permet d'accélérer le processus.

**Organisation de cette annexe.** Les travaux existants traitant de la capture de mouvement et du recalage d'objets déformables sont présentés en §A.2. La problématique est énoncée en §A.3. La procédure de minimisation utilisée pour minimiser la fonction de coût combinée est décrite en §A.4. Les résultats expérimentaux sont présentés en §A.5. Pour finir, nous donnons nos conclusions et discutons des travaux futurs en §A.6.

## A.2 Travaux antérieurs

Ces travaux sont liés à ceux traitant de la capture de déformations et du recalage de surfaces déformables.

**Capture de mouvement.** Plusieurs travaux ont été réalisés, notamment dans la communauté graphique, sur la modélisation des déformations de surfaces telles que les vêtements. Dans (Guskov et al., 2003) une texture spécifique constituée de différents carrés de couleur rouge, verte et bleue est suivie en utilisant un réseau de caméras. Un modèle de Markov est utilisé afin d'introduire des contraintes supplémentaires sur la consistance spatiale et sur l'apparence de la surface. Dans (Scholz et al., 2005) les auteurs proposent une texture spécifique basée sur la répétition de cinq cercles de couleur. Chaque groupement de  $3 \times 3$  cercles est unique, permettant la mise en correspondance de primitives entre les images issues de différentes caméras. Une estimation directe du champ de déplacement dense sur des images de profondeur est proposée dans (Yamamoto et al., 1990) ; des informations *a priori* complémentaires sont introduites afin de rendre le problème bien posé.

**Recalage d'objets déformables.** Le recalage de nuages de points 3D est une problématique complexe habituellement résolue en utilisant des algorithmes de type ICP dans le cas d'objets rigides. Deux étapes sont répétées jusqu'à convergence :

- ▷ Estimer les correspondances de points entre les différents nuages de points 3D, étant donnée la transformation.
- ▷ Estimer la transformation étant données les correspondances.

Les surfaces déformables ont récemment été traitées. Les approches existantes diffèrent principalement sur la transformation utilisée et dans la manière d'estimer les correspondances.

Sous sa forme d'origine, l'algorithme ICP (Besl and McKay, 1992) détermine les correspondances en recherchant les points les plus proches, introduisant une décision binaire. De nombreuses améliorations ont été proposées en relaxant cette décision binaire. Dans (Granger and Penneç, 2002), un poids est associé à chaque correspondance, il dépend de leur distance et d'un paramètre  $\sigma$  représentant le rayon d'une région de confiance : un poids négligeable est affecté aux correspondances présentant une distance bien supérieure à ce rayon. Pour un  $\sigma$  grand, l'algorithme introduit est très robuste mais pas très précis tandis que pour un  $\sigma$  petit les propriétés sont « inversées ». Une solution est alors d'utiliser une approche « multi-résolution » où le paramètre  $\sigma$  est diminué au cours des itérations de l'algorithme EM<sup>2</sup>. L'algorithme introduit est appelé EM-ICP. Une approche similaire est proposée dans (Chui and Rangarajan, 2003) ; les poids associés à chaque correspondance sont en plus normalisés afin d'imposer leur unicité. Enfin, (Fitzgibbon, 2003) met fin au schéma d'optimisation en deux étapes des approches décrites ci-dessus en utilisant un processus d'optimisation totalement non-linéaire via l'algorithme Levenberg-Marquardt (LM). Ce schéma d'optimisation permet d'inclure facilement un M-estimateur dans la fonction de coût. Il en résulte un algorithme, appelé LM-ICP, plus robuste que l'approche originelle (Besl and McKay, 1992).

La plupart des méthodes existantes utilisent des fonctions de type plaque mince pour représenter les déformations (Chui and Rangarajan, 2003; Jian and Vemuri, 2005) ou les défauts présent dans le nuage de points 3D (Brown and Rusinkiewicz, 2004). Dans (Myronenko et al., 2007), un champ de déplacement régularisé est utilisé. Les points proches dans l'espace sont supposés se déplacer de manière cohérente les uns par rapport aux autres. Une pénalité est proposée afin d'encourager des déformations « cohérentes ».

**L'approche proposée.** Notre algorithme s'appuie sur les points forts de certaines approches décrites précédemment. Il combine un modèle déformable représenté par un maillage régulier avec une

---

<sup>2</sup>"Expectation-maximization" en anglais.

procédure de recalage de type LM-ICP. Des contraintes de lissage spatio-temporelle ainsi que des informations sur les bords de la surface s'avèrent être indispensables pour empêcher la surface reconstruite de se comprimer ou de « glisser » arbitrairement. L'optimisation est réalisée de manière très efficace en utilisant la transformée en distance du nuage de points 3D.

## A.3 Énoncé du problème

### A.3.1 Représentation de la surface

Le capteur 3D fournit une séquence de nuages de points 3D  $Q_i$  :

$$Q_i \stackrel{\text{def}}{=} \begin{pmatrix} Q_{i,1}^x & \cdots & Q_{i,p}^x \\ Q_{i,1}^y & \cdots & Q_{i,p}^y \\ Q_{i,1}^z & \cdots & Q_{i,p}^z \end{pmatrix}. \quad (\text{A.1})$$

La surface  $S_i$  est représentée par  $m$  points 3D, traduisant les déformations d'une grille régulière ; elle est donnée par :

$$S_i \stackrel{\text{def}}{=} \begin{pmatrix} S_{i,1}^x & \cdots & S_{i,m}^x \\ S_{i,1}^y & \cdots & S_{i,m}^y \\ S_{i,1}^z & \cdots & S_{i,m}^z \end{pmatrix}. \quad (\text{A.2})$$

En pratique, le nombre de points 3D  $p$  est bien plus grand que le nombre de sommets  $m$  constituant le modèle. Notre algorithme détermine pour chacun des sommets si il existe une correspondance dans le nuage de points 3D et ceci en dépit de données manquantes ou erronées provenant principalement du bruit présent au niveau du capteur. La surface est finalement reconstruite en interpolant les sommets du maillage estimé.

### A.3.2 Fonction de coût

La fonction de coût combinée que nous proposons possède deux termes de données et trois termes de régularisation :

$$\mathcal{E}(S) \stackrel{\text{def}}{=} \mathcal{E}_{dg}(S) + \lambda_{db}\mathcal{E}_{db}(S) + \lambda_s\mathcal{E}_s(S) + \lambda_n\mathcal{E}_n(S) + \lambda_x\mathcal{E}_x(S), \quad (\text{A.3})$$

où  $\lambda_{db}$ ,  $\lambda_s$ ,  $\lambda_n$ ,  $\lambda_x$  sont des poids contrôlant l'influence des différents termes. Notons que les indices  $i$ , traduisant l'évolution temporelle des nuages de points 3D, sont enlevés par la suite pour des raisons de clarté. Par conséquent nous notons  $S \leftarrow S_i$  et  $\tilde{S} \leftarrow S_{i-1}$ .

Les termes de données servent à attirer le modèle sur le nuage de points 3D. Le premier d'entre eux  $\mathcal{E}_{dg}$  est un terme d'attraction globale tandis que le deuxième  $\mathcal{E}_{db}$  se limite aux bords de la surface. Leur robustesse aux données manquantes ou erronées est garantie par un estimateur robuste. Les termes de régularisation  $\mathcal{E}_s$  et  $\mathcal{E}_x$  encouragent respectivement des variations temporelles et spatiales lisses tandis que  $\mathcal{E}_n$  est une contrainte d'inextensibilité pénalisant la contraction et l'étirement du maillage. Les différents termes sont expliqués plus en détails ci-dessous.

**Le premier terme de données : l'attraction globale  $\mathcal{E}_{dg}$ .** Ce terme attire globalement le modèle sur les données 3D en se basant sur les distances entre les sommets du maillage et les points du nuage 3D les plus proches. Afin d'éviter le processus d'optimisation en deux étapes inhérent aux algorithmes de

type ICP, l'opérateur min est directement incorporé dans le terme d'attraction globale comme proposé dans (Fitzgibbon, 2003). Soit  $E_Q$  et  $E_S$  l'ensemble des points constituant respectivement les bords du nuage de points 3D et du modèle. Le terme d'attraction globale est alors donné par :

$$\sum_{\mathbf{S} \in S \setminus E_S} \min_{\mathbf{Q} \in Q \setminus E_Q} \|\mathbf{Q} - \mathbf{S}\|^2, \quad (\text{A.4})$$

où  $\mathbf{Q}$  et  $\mathbf{S}$  sont des vecteurs 3D représentant respectivement un point du nuage 3D et un point du modèle. Il est important de noter que contrairement à l'algorithme proposé dans (Fitzgibbon, 2003), les inconnues ne sont pas les paramètres d'une transformation rigide (rotation et translation) mais bien la position de chaque sommet du maillage. Les données aberrantes sont rejetées en introduisant un M-estimateur  $\rho$ . Les plus fréquemment utilisés dans la littérature sont présentés en §2.2.1.4. Le terme (A.4) robustifié est donné par :

$$\mathcal{E}_{dg}(\mathbf{S}) \stackrel{\text{def}}{=} \sum_{\mathbf{S} \in S \setminus E_S} \rho \left( \min_{\mathbf{Q} \in Q \setminus E_Q} \|\mathbf{Q} - \mathbf{S}\|^2 \right). \quad (\text{A.5})$$

**Le deuxième terme de données : l'attraction aux bords  $\mathcal{E}_{db}$ .** Ce terme attire les points appartenant aux bords du modèle sur ceux constituant les bords du nuage de points 3D. Il est semblable au terme d'attraction global, excepté que la somme et l'opérateur min sont définis uniquement sur les points appartenant aux bords :

$$\mathcal{E}_{db}(\mathbf{S}) \stackrel{\text{def}}{=} \sum_{\mathbf{S} \in E_S} \rho \left( \min_{\mathbf{Q} \in E_Q} \|\mathbf{Q} - \mathbf{S}\|^2 \right). \quad (\text{A.6})$$

**Terme de régularisation : lissage spatial  $\mathcal{E}_s$ .** Ce terme favorise les déformations lisses plus à même de représenter des déformations plausibles de la surface observée. Nous utilisons la forme discrète de l'énergie de courbure étendue au cas 3D :

$$\mathcal{E}_s^2(\mathbf{S}) \stackrel{\text{def}}{=} \text{vect}(\mathbf{S})^\top \mathcal{Y} \text{vect}(\mathbf{S}), \quad (\text{A.7})$$

avec  $\mathcal{Y} \stackrel{\text{def}}{=} \text{diag}(\mathcal{C}, \mathcal{C}, \mathcal{C})$ ,  $\mathcal{C}$  est la matrice de taille  $m \times m$  représentant l'énergie de courbure discrétisée dont la définition est donnée en §2.2.1.3.

**Terme de régularisation : lissage temporel  $\mathcal{E}_x$ .** Ce terme définit une dépendance entre les surfaces aux instants  $i$  et  $i - 1$ ,  $\mathbf{S}$  et  $\tilde{\mathbf{S}}$  respectivement :

$$\mathcal{E}_x(\mathbf{S}) \stackrel{\text{def}}{=} \|\mathbf{S} - \tilde{\mathbf{S}}\|^2. \quad (\text{A.8})$$

Des déformations temporellement lisses sont alors favorisées, ce qui est une hypothèse relativement raisonnable lorsque des séquences sont traitées.

**Terme de régularisation : contrainte d'inextensibilité  $\mathcal{E}_n$ .** Ce terme est décrit précédemment en §5.2.1.2. Nous rappelons ici sa définition :

$$\mathcal{E}_n(\mathbf{S}) \stackrel{\text{def}}{=} \sum_{\mathbf{S} \in S} \sum_{\hat{\mathbf{S}} \in \mathbf{N}(\mathbf{S})} \left( \|\mathbf{S} - \hat{\mathbf{S}}\|^2 - L_{s,\hat{s}}^2 \right)^2, \quad (\text{A.9})$$

avec  $N(S)$  l'ensemble des sommets voisins de  $S$  et  $L_{s,\hat{s}}$  la distance sur le maillage de référence (avant déformation) entre les sommets  $S$  et  $\hat{S}$ . Ce terme pénalise l'étirement et la contraction du maillage poussant les distances entre sommets voisins à rester constantes.

## A.4 Procédure de minimisation

La fonction de coût combinée (A.3) est une somme de résidus au carré dépendant non-linéairement des inconnues dans  $S$ . Elle est minimisée par l'algorithme Levenberg-Marquardt décrit en §1.6.2. Plus précisément nous avons étendu l'algorithme LM-ICP proposé dans (Fitzgibbon, 2003) aux surfaces déformables. L'approximation de la matrice Hessienne  $H = \mathcal{J}^T \mathcal{J} + \tau I$  doit être inversée à chaque itération ce qui peut être coûteux en temps de calcul si le nombre de sommets constituant le modèle est trop important. La matrice Jacobienne  $\mathcal{J}$  rassemble celles associées à chaque terme de la fonction de coût :

$$\mathcal{J} = \begin{pmatrix} \mathcal{J}_{dg} & \mathcal{J}_{db} & \mathcal{J}_s & \mathcal{J}_n & \mathcal{J}_x \end{pmatrix}, \quad (\text{A.10})$$

où  $\mathcal{J}_{dg}$ ,  $\mathcal{J}_{db}$ ,  $\mathcal{J}_s$ ,  $\mathcal{J}_n$ ,  $\mathcal{J}_x$  sont respectivement les matrices Jacobiennes associées aux termes d'attraction global, d'attraction aux bords, de lissage spatial, d'inextensibilité et de lissage temporel. Elles ont pour tailles respectives  $(m - \text{card}(E_S)) \times 3m$ ,  $\text{card}(E_S) \times 3m$ ,  $3m \times 3m$ ,  $\text{card}(N(S)) \times 3m$  et  $m \times 3m$ , où  $\text{card}(A)$  est le cardinal de  $A$  et  $N(S)$  est l'ensemble des connections entre sommets voisins de la surface  $S$ . Lors de nos expérimentations, une grille de  $15 \times 20$  sommets est utilisée ce qui introduit une matrice Jacobienne constituée de  $3.694 \times 900$  éléments ( $m = 300$ ,  $\text{card}(E_S) = 66$ ,  $\text{card}(N(S)) = 2.194$ ). L'estimation des points du nuage 3D les plus proches de ceux du modèle peut être résolue efficacement en utilisant une transformée en distance, comme suggéré dans (Fitzgibbon, 2003). Le principe de la transformée en distance est décrit en §1.3. Une carte de distance volumétrique est pré-calculée entre le nuage de points 3D et son volume englobant discrétisé. L'estimation des distances aux points les plus proches revient alors à interpoler la carte volumétrique. Les gradients de la carte volumétrique (obtenus par différence finie) sont utilisés pour estimer la matrice Jacobienne  $\mathcal{J}_{dg}$ . Ils présentent l'avantage d'être constants au cours des itérations. La figure A.1 illustre le principe de la transformée en distance sur un nuage de points 2D.

La matrice Jacobienne  $\mathcal{J}$  de la fonction de coût combinée est extrêmement creuse. Cette propriété est prise en compte afin d'accélérer le processus d'optimisation. Nous utilisons le module de factorisation de Cholesky directement disponible dans le logiciel Matlab<sup>3</sup>. La figure A.2 illustre la nature creuse de la matrice Jacobienne et de la matrice Hessienne.

## A.5 Résultats expérimentaux

Nous avons testé l'approche proposée sur deux types d'expériences : le premier jeu de données<sup>4</sup> provient d'un scanner 3D à lumière structurée observant les déformations d'une feuille de papier. Pour la deuxième expérience, le capteur est un système stéréo<sup>5</sup> permettant l'acquisition de nuages de points 3D en temps réel ; les déformations d'une couverture sont observées avec ce capteur. La condition initiale détermine à la fois la position du modèle ainsi que la taille de la grille. Elle se doit d'être

<sup>3</sup>Plus précisément, nous utilisons la fonction "mldivide" de Matlab.

<sup>4</sup>Données fournies par Johnny Park (<http://web.ics.purdue.edu>).

<sup>5</sup>Ce capteur provient de la société eVS à Vérone (<http://www.evsys.net>).

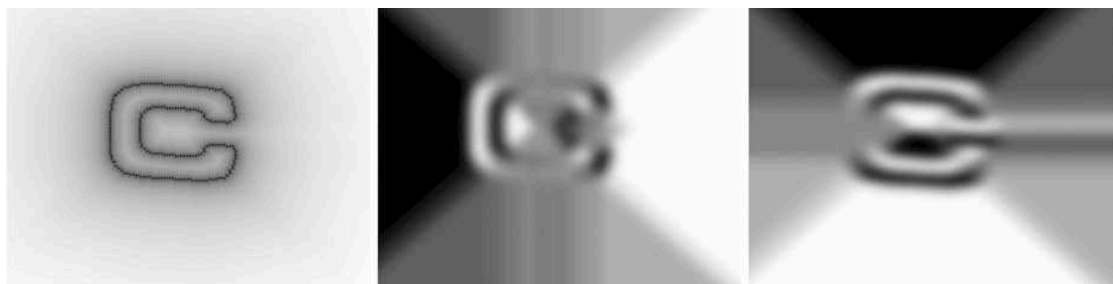


FIG. A.1 – **Transformée en distance sur un nuage de points 2D**. A gauche : la carte de distance. En noir, en blanc, les régions de l'espace pour lesquelles la distance au nuage de points 2D est respectivement faible, élevée. Au centre : le gradient de la carte de distance suivant  $x$ . A droite : le gradient de la carte de distance suivant  $y$ . Images extraites de (Fitzgibbon, 2003).

proche de la solution optimale afin de garantir la bonne convergence de l'algorithme LM-ICP et de déterminer précisément les distances  $L_{s,\hat{s}}$  de l'équation (A.9). La détection des points appartenant aux bords n'est pas l'objectif ici. L'approche proposée étant robuste aux erreurs de mises en correspondance, des opérateurs simplistes de détection des bords sont utilisés. En l'absence de texture comme c'est le cas pour la première expérience, ils sont détectés en recherchant les fortes discontinuités de profondeur dans le nuage de points 3D. Pour la deuxième expérience, les images d'intensité sont utilisées pour segmenter les contours. Le résultat est assez bruité mais ne perturbe pas trop la qualité de la reconstruction.

### A.5.1 Une feuille de papier observée par un scanner à lumière structurée

Ce capteur fournit un nuage de points 3D précis et haute résolution. Différentes déformations d'une feuille de papier ne présentant pas de consistance temporelle sont capturées. Le terme de lissage temporelle  $\mathcal{E}_x$  est donc désactivé pour cette expérience. L'orientation initiale de la grille est estimée en recalant un plan sur les données. La projection des points 3D sur ce plan définit la taille et les bords de la grille. Les figures A.3(a), A.3(b) et A.3(c) présentent trois exemples de nuages de points 3D. La première ligne illustre les données d'entrée et la position initiale du modèle. Pour le premier exemple A.3(a), les déformations sont situées principalement le long des contours horizontaux, pour le deuxième exemple A.3(b) le papier est plié en rabattant le coin supérieur droit vers le coin inférieur gauche ; enfin, pour le troisième exemple A.3(c) les déformations sont réparties un peu partout sur la surface. Les résultats du recalage sont présentés sur la deuxième ligne des figures A.3(a), A.3(b) et A.3(c). Les déformations estimées sont précises aussi bien pour les points intérieurs du maillage que pour ceux appartenant aux bords, tout en étant lisses comme espéré. Des images synthétiques ont été générées à partir des surfaces reconstruites ; elles sont représentées sur la dernière ligne des figures A.3(a), A.3(b) et A.3(c). Le rendu obtenu est très réaliste quelque soit la texture utilisée, comme l'illustre la figure A.4.

La même grille est utilisée pour chaque nuage de points. Les correspondances entre les différentes reconstructions sont alors naturellement connues : les points en correspondance sont ceux situés à la même position sur la grille. Des déformations intermédiaires peuvent être générées synthétiquement en interpolant la trajectoire des sommets entre différentes reconstructions. La figure A.5 montre des

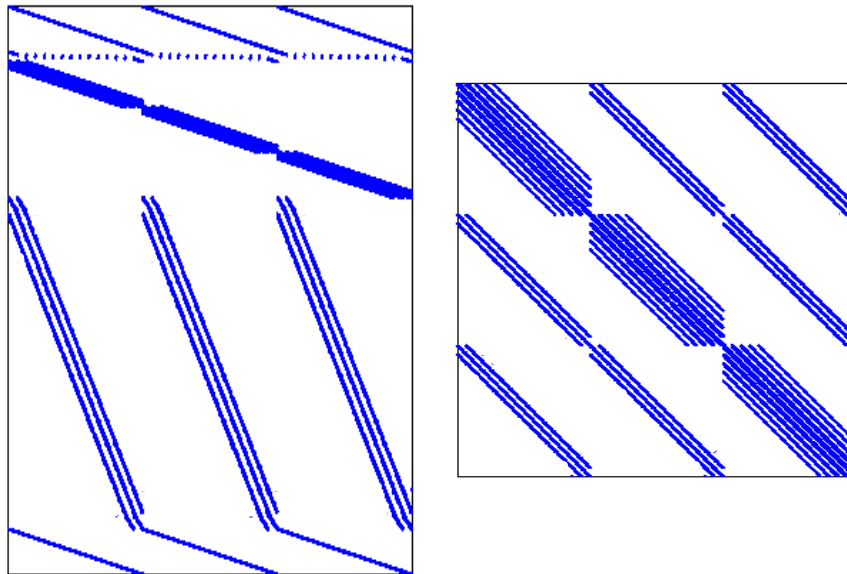


FIG. A.2 – **Structures des matrices Jacobienne et Hessienne.** A gauche : structure de la matrice Jacobienne. Sa taille est de  $3.694 \times 900$ , elle a 3.324.600 composantes dont seulement 25.626 sont non-nulles. A droite : structure de la matrice Hessienne. Sa taille est de  $900 \times 900$ , elle a 810.000 composantes dont seulement 46.716 sont non-nulles.

exemples de déformations intermédiaires obtenues par une simple interpolation bilinéaire entre les sommets des modèles évalués sur les données représentées sur les figures A.3(a), A.3(b), A.3(c). Un schéma d'interpolation plus complexe préservant les propriétés physiques de la surface observée est à privilégier. Ce dernier devra notamment incorporer la contrainte d'inextensibilité.

## A.5.2 Une couverture observée par un système stéréoscopique

Ce capteur acquiert des images en temps réel (25 images par seconde) et fournit en sortie des informations 3D et d'intensité. Une longue séquence, présentant des déformations d'une couverture, est capturée : la qualité des images est faible. De plus, les nuages de points 3D sont bruités. Notons également que le champ de vue de ce capteur est très limité, environ  $30\text{cm}^3$ .

Une image de la couverture est représentée sur la figure A.6(a). Seule la région délimitée par le carré noir est prise en compte lors de cette expérience. Notons que seules les correspondances 2D entre l'image de gauche et celle de droite sont estimées précisément par le système stéréo. Les correspondances temporelles ne sont pas fiables en raison des distortions introduites par les déformations de la couverture. La figure A.6(b) représente le nuage de points 3D fourni par le système stéréo pour la première image de la séquence : un grand nombre de points 3D sont aberrants notamment sur les bords. La scène est en outre difficilement reconnaissable.

Nous utilisons les images d'intensité afin de sélectionner automatiquement la région d'intérêt, c'est-à-dire le carré noir, dont les contours sont extraits par des techniques standards d'analyse d'images. La figure A.7(a) illustre les contours extraits de l'image d'intensité (figure A.6(a)) tandis que la figure A.7(b) représente les données 3D sélectionnées (nuage de points 3D et points appartenant aux bords), c'est-à-dire celles relatives à la région d'intérêt.

Le modèle est dans un premier temps initialisé sur le premier nuage de points 3D en recalant un



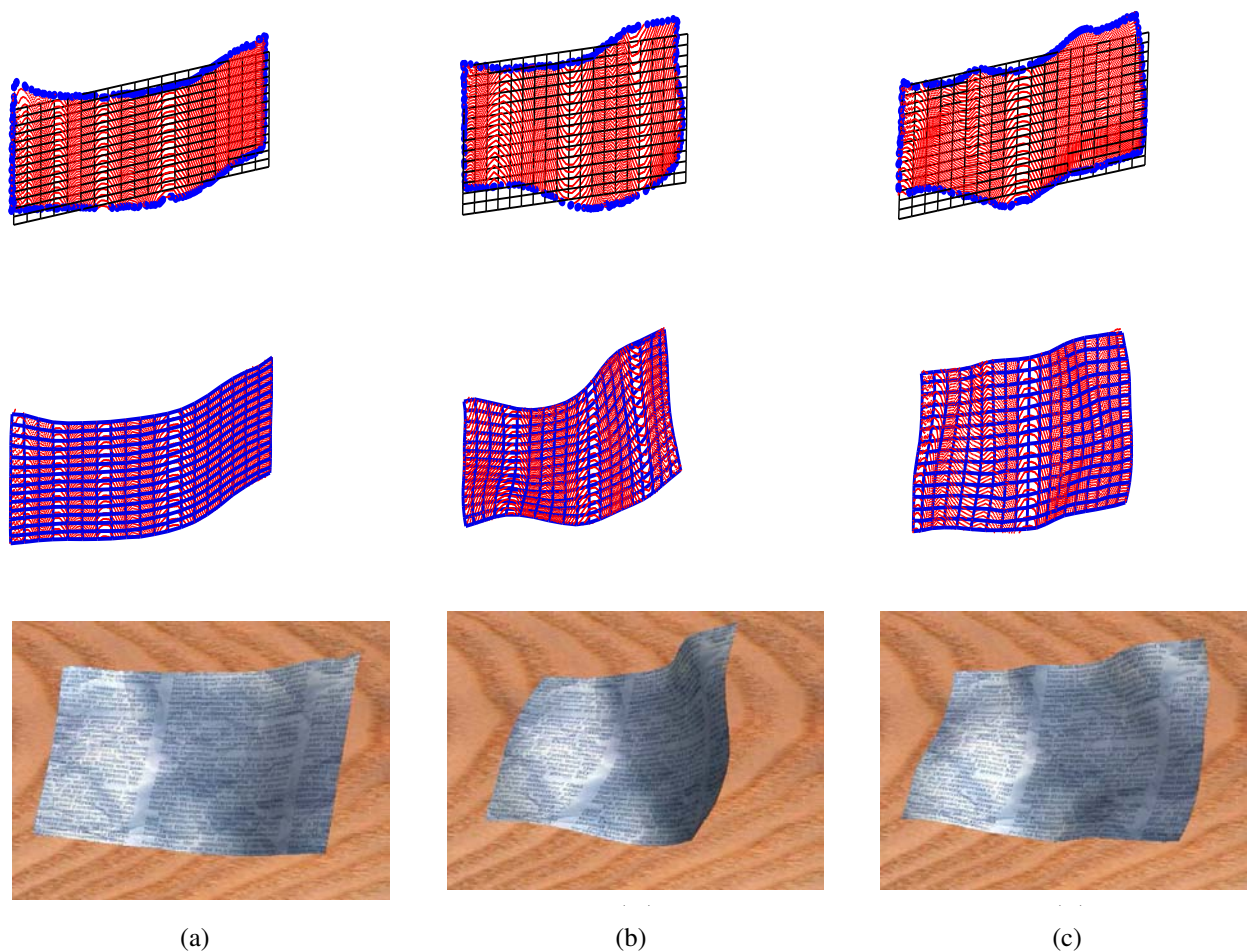


FIG. A.3 – **Résultat du recalage sur trois nuages de points 3D provenant d’un scanner à lumière structurée.** En haut : positions initiales du maillage. Au centre : recalage du modèle. En bas : images synthétiques générées à partir des surfaces reconstruites.

plan, ensuite la condition initiale pour les données courantes est la position du maillage sur le nuage de points précédents. Les déformations estimées au cours de cette séquence sont illustrées sur la figure A.8. Pour chaque exemple on visualise :

- ▷ L’image d’intensité avec les contours extraits et la reprojection du modèle estimé.
- ▷ Le nuage de points 3D avec le modèle 3D estimé. Les points 3D détectés comme appartenant aux bords de la région d’intérêt sont mis en évidence.

L’information d’intensité est utilisée ici uniquement pour définir la région d’intérêt ; elle n’est pas prise en compte pour évaluer les déformations du modèle.

La couverture est manipulée à partir des coins inférieur gauche et supérieur droit. Pour les premières images de la séquence, le coin supérieur droit de la couverture est progressivement rabattu en direction du centre de la région d’intérêt. Ensuite celle-ci est fortement étirée afin d’éloigner le plus possible les coins supérieur droit et inférieur gauche. Pour finir, des déformations aléatoires sont générées, notamment dans les différents coins. Certaines données sont très difficiles de par la présence de fortes déformations, de bruit et de trous dans les données et dans la détection des bords. Par exemple, sur l’image (c) le coin supérieur droit est fortement déformé, sur l’image (f) la détection des contours est erronée et présente un trou dans le coin supérieur droit. Enfin, sur les images (h) et



FIG. A.4 – Images synthétiques générées avec différentes textures.

(i), les données 3D au niveau du bord inférieur gauche sont clairement aberrantes. Les résultats sont néanmoins satisfaisants puisque les déformations estimées sont correctes tout en étant lisses. Leur projection sur les images d'intensité corrobore la précision de l'approche proposée.

Nous utilisons les déformations estimées au cours de cette séquence pour générer synthétiquement une vidéo représentant un film projeté sur un écran déformable. Des images extraites de cette vidéo sont représentées sur la figure A.9.

### A.5.3 Discussion

Pour les deux expériences, un maillage de taille  $15 \times 20$  est utilisé. Nous avons observé expérimentalement qu'un poids plus important doit être accordé au terme d'attraction aux bords :  $\lambda_b = 1,5$ . Ceci permet de garantir la convergence de l'algorithme vers une solution proche de la réalité, les autres paramètres sont fixés à 1 lors des différentes expériences. Ces valeurs donnent de bons résultats sur la plupart des données que nous avons testées. Néanmoins, un choix automatique des paramètres doit être envisagé lors de travaux futurs.

Le tableau A.1 récapitule les temps de calcul requis pour estimer les déformations du maillage pour les deux expériences présentées ci-dessus. Nous mesurons le coût des prétraitements (initialisation de la grille, calcul des  $L_{s,\hat{s}}$  et de la transformée en distance), le coût moyen d'une itération, le nombre moyen d'itérations nécessaires pour converger, le coût moyen pour traiter un nuage de points 3D, le coût total pour traiter la séquence.

Le coût d'une itération de LM est semblable pour les deux expériences. Il dépend uniquement de la taille du maillage représentant le modèle et non pas du nombre de points constituant le nuage 3D. En revanche, le temps passé pour les prétraitements varie de par l'évaluation de la transformée en distance. Cette dernière dépend en partie du nombre de points constituant le nuage 3D. L'algorithme

Expérience / nb. nuages 3D	Prétraitements (s)	Coût Itér. (s)	Nb. itér.	Un nuage (s)	Séquence (s)
Feuille de papier / 1	19.36	2.22	11	24.42	43.78
Couverture / 100	15.24	2.34	7	16.34	1649.34

TAB. A.1 – Temps de calcul requis pour reconstruire les déformations d’une surface avec l’algorithme proposé.

est implanté en Matlab sur un Pentium 4 1,86GHz.

## A.6 Conclusion

Nous avons proposé une nouvelle approche pour capturer les déformations d’une surface à partir de données issues d’un capteur 3D. Le modèle représentant la surface est un maillage régulier, déformé sous différentes contraintes, afin de correspondre parfaitement aux nuages de points 3D. La fonction de coût introduite est composée d’informations *a priori*, comme par exemple un lissage spatio-temporel des déformations et de termes d’attraction aux données. Les termes d’inextensibilité et d’attraction aux bords apparaissent être cruciaux pour rendre bien posé ce problème naturellement ambigu. La fonction de coût est minimisée par l’algorithme LM-ICP en prenant en compte la nature extrêmement creuse des matrices (Hessienne et Jacobienne) mises en jeu. Les résultats obtenus sont très prometteurs ; les performances de reconstruction sont satisfaisantes sur les cas traités. L’approche proposée est testée sur deux jeux de données issus de capteurs différents, démontrant sa généralité. L’algorithme proposé est en outre relativement rapide et est très facile à implanter.

Nous avons appliqué notre approche à la synthèse d’images et de vidéos présentant une surface déformable. D’autres applications peuvent néanmoins être visées, comme la réalité augmentée, notamment pour les effets spéciaux, ou bien la compression de données.

En perspective, la sélection automatique des poids associés aux différents termes constituant la fonction de coût devra être étudiée. Nous avons déjà discuté de cette problématique dans la conclusion de ce manuscrit.

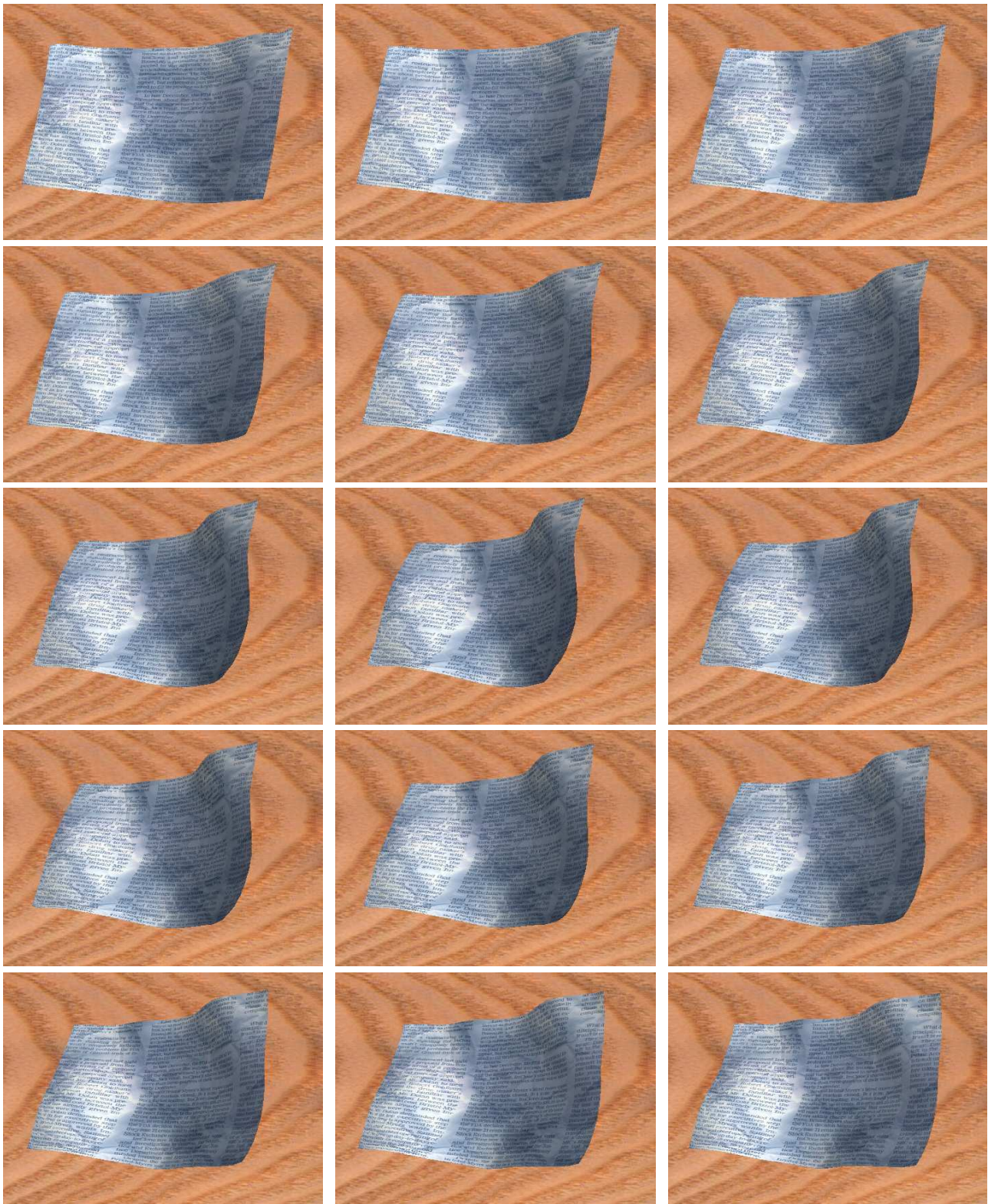


FIG. A.5 – Images synthétisées en interpolant les surfaces reconstruites à partir des données des figures A.3(a), A.3(b) et A.3(c).

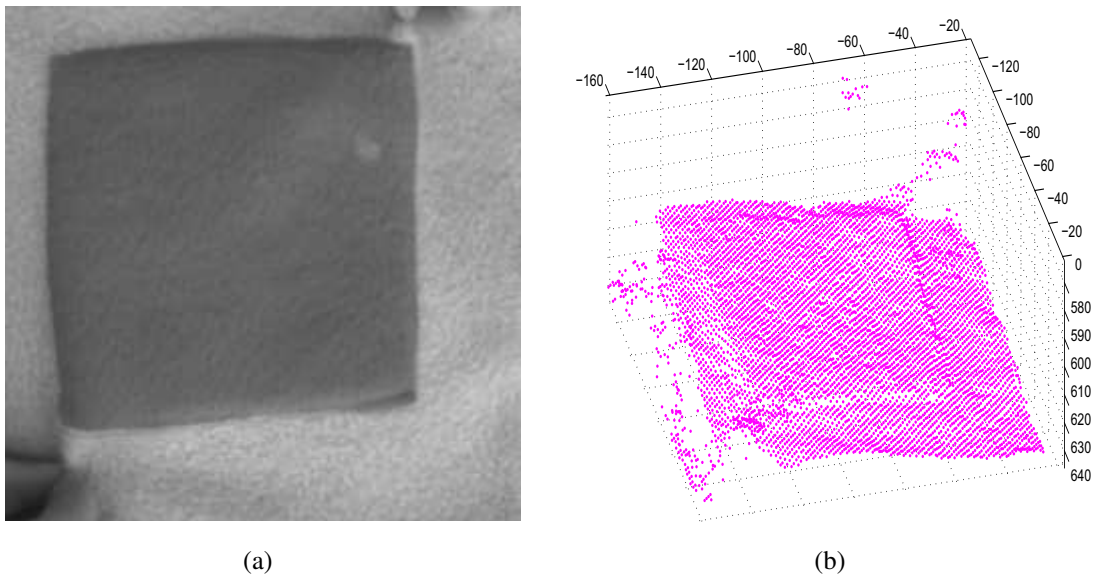


FIG. A.6 – **La séquence de la couverture.** (a) Image d'intensité représentant la couverture. (b) Le nuage de points 3D associé.

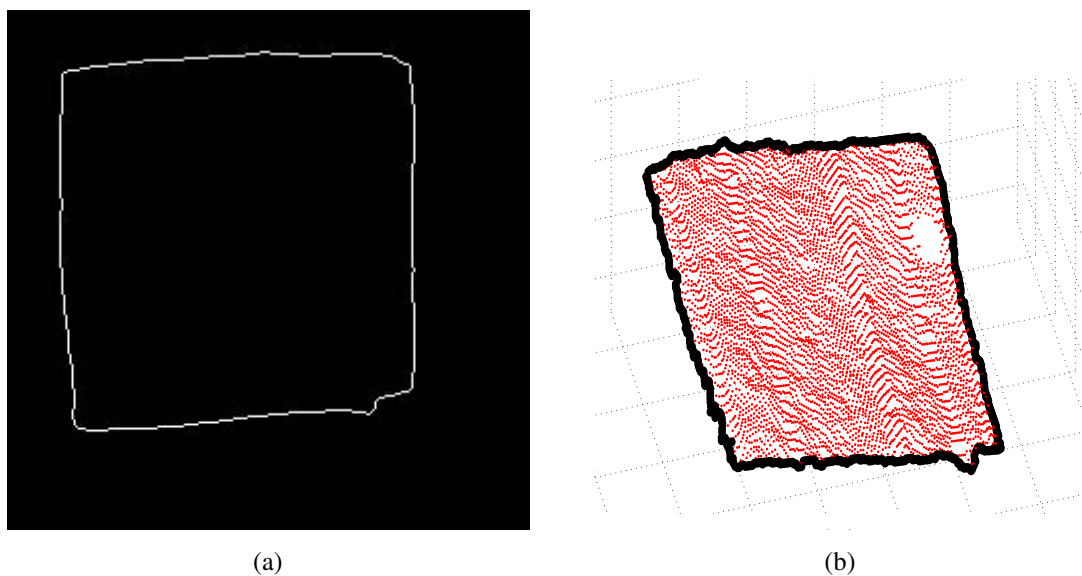


FIG. A.7 – **Extraction des données d'entrée pour la séquence de la couverture.** (a) Les contours 2D. (b) Le nuage de points 3D correspondant uniquement à la région d'intérêt. Les points 3D appartenant aux bords de la région d'intérêt sont représentés en noir.

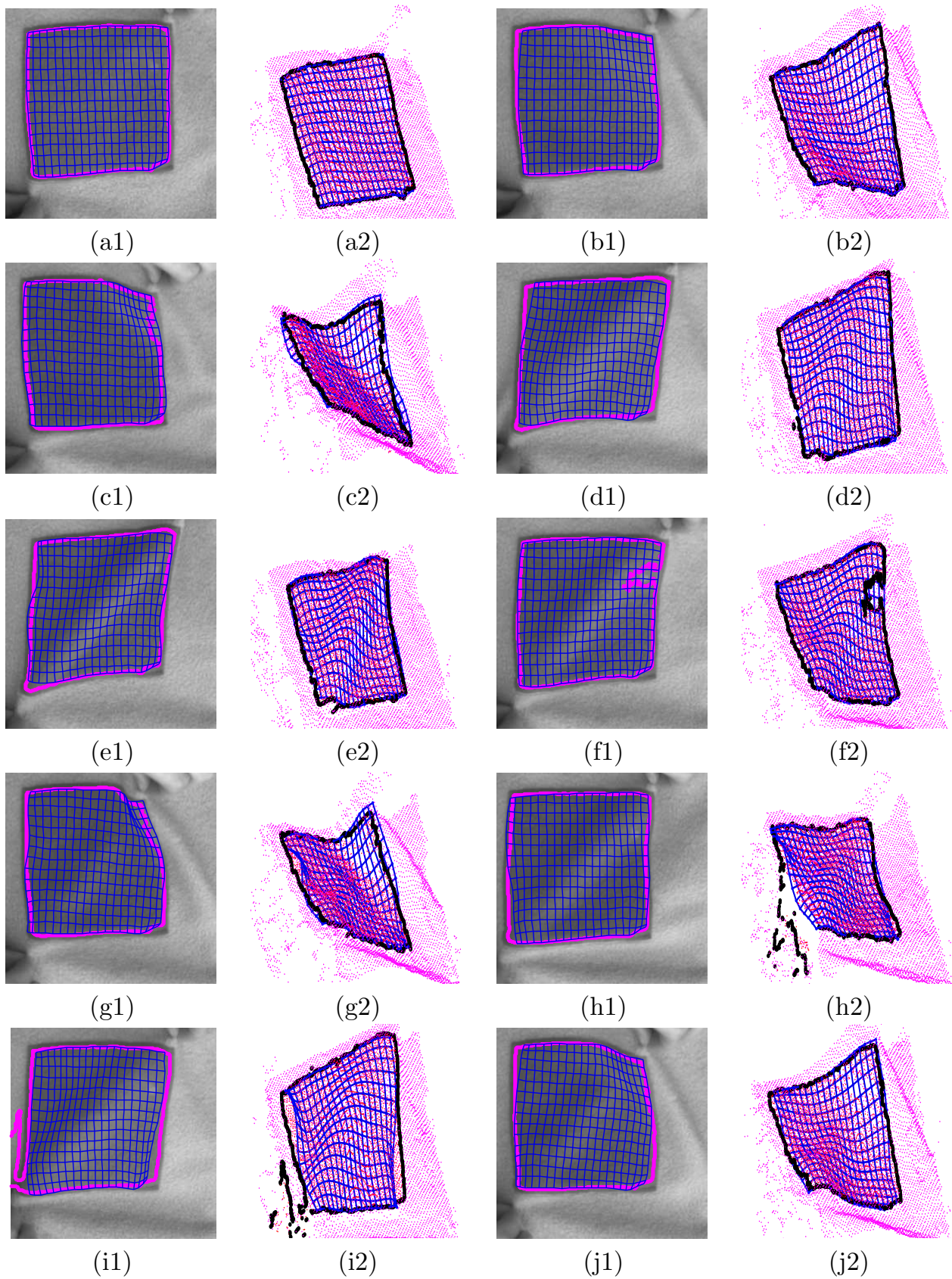


FIG. A.8 – **Résultat du recalage sur la séquence de la couverture.** (·1) les images d'intensité sur lesquelles sont surimposés les contours 2D ainsi que la reprojection des modèles estimés. (·2) les nuages de points 3D et les modèles 3D estimés.

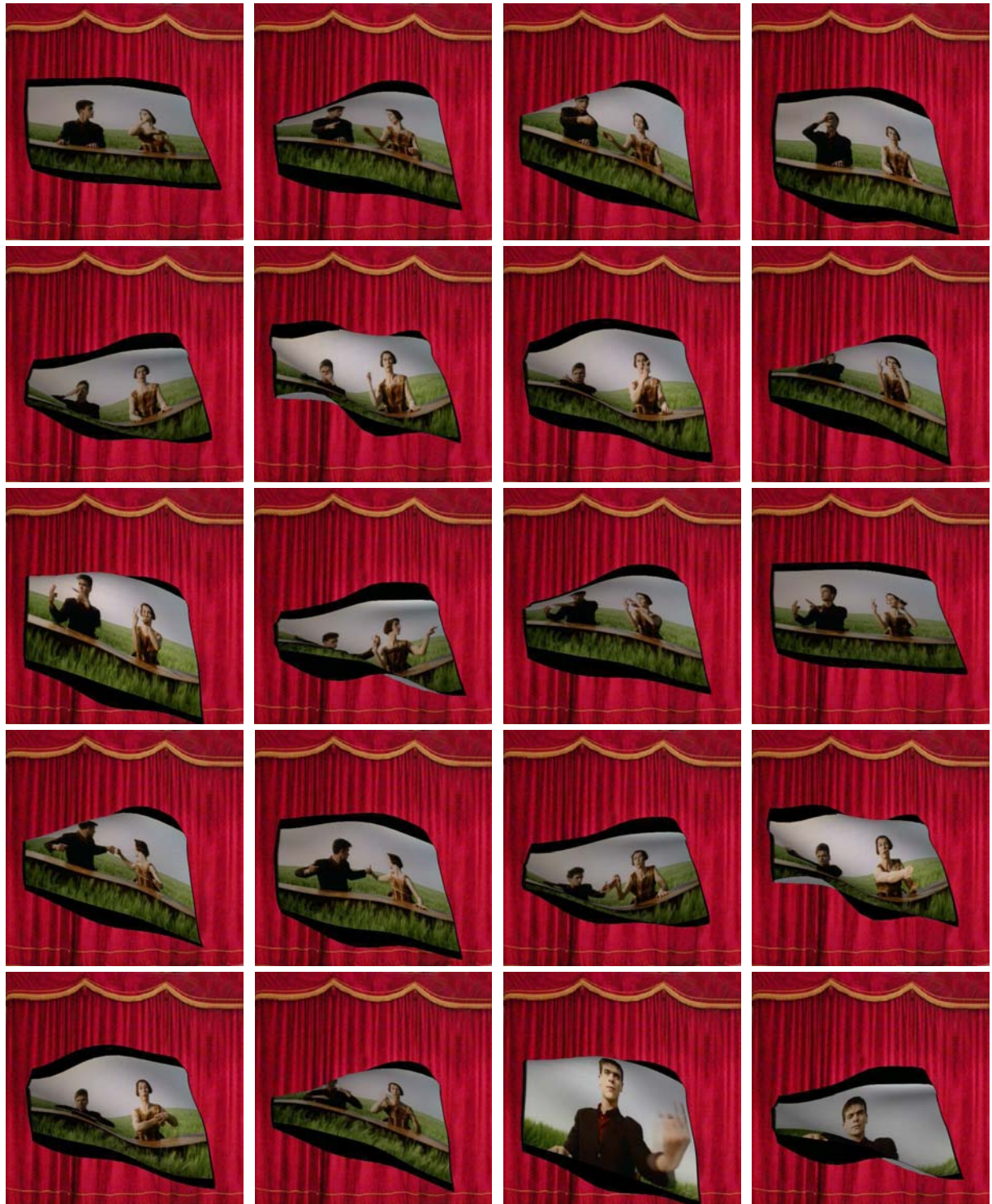


FIG. A.9 – Vidéo de synthèse représentant un film sur un écran déformable. Chaque image constituant le film est projetée sur la surface reconstruite au cours de la séquence de la couverture.





# Bibliographie

- H. Aanæs and F. Kahl. Estimation of deformable structure and motion. In *Proceedings of the Vision and Modelling of Dynamic Scenes Workshop*, 2002.
- A. Agarwal and B. Triggs. Recovering 3D human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1) :44–58, January 2006.
- J. Ahlberg. Candide 3 : an Updated Parameterised Face. Technical report, Dept. of Electrical Engineering, Linköping University, Sweden, 2001.
- N. Arad and D. Reissfeld. Image warping using few anchor points and radial functions. In *Computer Graphics Forum*, 1995.
- S. Baker and I. Matthews. Lucas-Kanade 20 years on : A unifying framework. *International Journal of Computer Vision*, 56(3) :221–255, February 2004.
- A. Bartoli. Groupwise geometric and photometric direct image registration. In *Proceedings of the British Machine Vision Conference*, 2006.
- A. Bartoli. Maximizing the predictivity of smooth deformable image warps through cross-validation. *Mathematical Imaging and Vision*, 31(2-3) :133–145, February 2007.
- A. Bartoli. *Contributions to Image Registration and to the 3D Reconstruction of Rigid and Deformable Scenes*. Habilitation à diriger des recherches, Université de Clermont-Ferrand II, 2008.
- A. Bartoli and A. Zisserman. Direct estimation of non-rigid registrations. In *Proceedings of the British Machine Vision Conference*, 2004.
- A. Bartoli, M. Perriollat, and S. Chambon. Generalized thin-plate spline warps. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.
- A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.
- H. Bay, T. Tuytelaars, and L. J. Van Gool. SURF : Speeded up robust features. In *Proceedings of the European Conference on Computer Vision*, 2006.
- S. Benhimane and E. Malis. Real-time image-based tracking of planes using efficient second-order minimization. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004.

- P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2) :239–256, February 1992.
- V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of SIGGRAPH*, 1999.
- F. L. Bookstein. Principal warps : Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6) :567–585, June 1989.
- G. Borgefors. Hierarchical chamfer matching : a parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6) :849–865, 1988.
- T. E. Boult and L. G. Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE Workshop on Motion Understanding*, 1991.
- D. Bradley and G. Roth. Augmenting non-rigid objects with realistic lighting. Technical Report NRC/ERB-1116, National Research Council of Canada, October 2004.
- M. Brand. Morphable 3D models from video. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001.
- M. Brand. A direct method for 3D factorization of nonrigid motion observed in 2D. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- M. Brand and R. Bhotika. Flexible flow for 3D nonrigid tracking and shape recovery. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001.
- C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2000.
- B. J. Brown and S. Rusinkiewicz. Non-rigid range-scan alignment using thin-plate splines. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2004.
- F. Brunet, A. Bartoli, R. Malgouyres, and N. Navab. L-tangent norm : A low computational cost criterion for choosing regularization weights and its use for range surface reconstruction. In *Proceedings of the International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.
- A. Buchanan and A. W. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31(4) :532–540, 1983.
- U. Castellani, V. Gay-Bellile, and A. Bartoli. Joint reconstruction and registration of a deformable planar surface observed by a 3D sensor. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, 2007.

- U. Castellani, V. Gay-Bellile, and A. Bartoli. Robust deformation capture from temporal range data for surface rendering. *Computer Animation and Virtual Worlds*, To appear.
- T. Chateau, J-T. Lapresté, D. Ramadasan, and S. Treuillet. Suivi de motifs planaires temps réel par combinaison de traqueurs. In *Proceedings of Reconnaissance des Formes et Intelligence Artificielle*, 2008.
- M. Chaumont and B. Beaumesnil. Robust and real-time 3D-face model extraction. In *Proceedings of the International Conference on Image Processing*, 2005.
- Y. Chen and F. Davoine. Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically. In *Proceedings of the British Machine Vision Conference*, 2006.
- H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2) :114–141, February 2003.
- T. Cootes and C. Taylor. Statistical models of appearance for medical image analysis and computer vision. *SPIE Medical Imaging*, 4322 :236–248, July 2001.
- T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, 1998.
- A. Del Bue. A factorization approach to structure from motion with shape priors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.
- A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using non-parametric tracking and non-linear optimization. In *Proceedings of the IEEE Workshop on Articulated and Non-Rigid Motion*, 2004.
- A. Del Bue, X. Llado, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.
- F. Dornaika and F. Davoine. Simultaneous facial action tracking and expression recognition using a particle filter. In *Proceedings of the International Conference on Computer Vision*, 2005.
- O. Faugeras. *Three-Dimensional Computer Vision : a Geometric Viewpoint*. MIT press, 1993.
- M. A. Fischler and R. C. Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) : 726–740, June 1981.
- A. W. Fitzgibbon. Robust registration of 2D and 3D point sets. *Image and Vision Computing Journal*, 21(13-14) :1145–1153, December 2003.
- A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *Proceedings of the European Conference on Computer Vision*, 1998.

- M. Fornefett, K. Rohr, and H. S. Stiehl. Elastic registration of medical images using radial basis functions with compact support. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1999.
- V. Gay-Bellile, M. Perriollat, A. Bartoli, and P. Sayd. Image registration by combining thin-plate splines with a 3D morphable model. In *Proceedings of the International Conference on Image Processing*, 2006.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Feature-driven non-rigid image registration. In *Proceedings of the British Machine Vision Conference*, 2007a.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning. In *Proceedings of the International Conference on Computer Vision*, 2007b.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Gestion des auto-occultations pour l'augmentation d'une surface déformable sur une vidéo. In *Proceedings of COMPRESSION et REPRÉSENTATION DES SIGNAUX AUDIOVISUELS*, 2007c.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Deformable surface augmentation in spite of self-occlusions. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2007d.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Estimation directe d'alignements non-rigides guidés par primitives. In *Proceedings of onzième congrès francophone des jeunes chercheurs en vision par ordinateur*, 2007e.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Recalage non-rigide direct avec prise en compte des auto-occultations au niveau image. In *Proceedings of Reconnaissance des Formes et Intelligence Artificielle*, 2008.
- V. Gay-Bellile, A. Bartoli, and P. Sayd. Direct estimation of non-rigid registrations with image-based self-occlusion reasoning (révision majeure effectuée). In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Soumis en 2008.
- S. Granger and X. Pennec. Multi-scale EM-ICP : A fast and robust approach for surface registration. In *Proceedings of the European Conference on Computer Vision*, 2002.
- N. Guilbert, A. Bartoli, and A. Heyden. Affine approximation for direct batch recovery of euclidean motion from sparse data. *International Journal of Computer Vision*, 69(3) :317 – 333, September 2006.
- I. Guskov, S. Klivanov, and B. Bryant. Trackable surfaces. In *Proceedings of Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003.
- G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10) :1025–1039, October 1998.
- F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust statistics : the approach based on influence functions*. Wiley, 1986.

- P. Hansen. The L-curve and its use in the numerical treatment of inverse problems. Technical report, University of Denmark, 2005.
- R. I. Hartley. In defence of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6) :580–593, June 1997.
- R. I. Hartley and F. Schaffalitzky. Powerfactorization : an approach to affine reconstruction with missing and uncertain data. In *Proceedings of the Australia-Japan Advanced Workshop on Computer Vision*, 2003.
- R. I. Hartley and P. F. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2) : 146–157, 1997.
- R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition.
- A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Proceedings of the Scandinavian Conference on Image Analysis*, 1997.
- A. Hilsmann and P. Eisert. Tracking deformable surfaces with optical flow in the presence of self occlusions in monocular image sequences. In *Proceedings of the NORDIA Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment at CVPR*, 2008.
- K. P. Horn and G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.
- T. S. Huang and O. D. Faugeras. Some properties of the E-matrix in two view motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(12) :1310–1312, December 1989.
- S. Ilic, M. Salzmann, and P. Fua. Implicit meshes for effective silhouette handling. *International Journal of Computer Vision*, 72(2) :159–178, April 2007. ISSN 0920-5691.
- M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proceedings of the International Conference on Computer Vision*, 1999.
- B. Jian and B.C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- H. J. Johnson and G. E. Christensen. Consistent landmark and intensity-based image registration. *IEEE Transactions on Medical Imaging*, 21(5) :450–461, May 2002.
- F. Jurie and M. Dhome. Hyperplane approximation for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :996–1000, July 2002a.
- F. Jurie and M. Dhome. Real time robust template matching. In *Proceedings of the British Machine Vision Conference*, 2002b.
- A. Ladikos, S. Benhimane, and N. Navab. A real-time tracking system combining template-based and feature-based approaches. In *International Conference on Computer Vision Theory and Applications*, 2007.

- V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects : A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1) :1–89, August 2005.
- V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2004.
- V. Lepetit, P. Lager, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.*, 2 :164–168, 1944.
- J. Lim and M.-H. Yang. A direct method for non-rigid motion with thin-plate splines. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- W.-C. Lin and Y. Liu. Tracking dynamic near-regular texture under occlusion and rapid movements. In *Proceedings of the European Conference on Computer Vision*, 2006.
- D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, November 2004.
- E. Malis. Improving vision based control using efficient second-order minimization techniques. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2004.
- E. Malis and E. Marchand. Experiments with robust estimation techniques in real-time robot vision. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2006.
- D. Marquardt. An algorithm for least-squares estimation of non linear parameters. *J. Soc. Industr. Appl. Math.*, 11(1) :431–444, 1963.
- D. Martinec and T. Pajdla. 3D reconstruction by fitting low-rank matrices with missing data. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2) :135–164, November 2004.
- K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10) :1615–1630, October 2005.
- A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration : Coherent point drift. In *Neural Information Processing Systems*, 2007.
- S. Olsen and A. Bartoli. Implicit non-rigid structure-from-motion with priors. *Journal of Mathematical Imaging and Vision, special issue : tribute to Peter Johansen*, 31(2-3) :233–244, July 2008.
- M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.
- M. Perriollat and A. Bartoli. A quasi-minimal model for paper-like surfaces. In *Proceedings of the Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images at CVPR*, 2007.

- J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
- J. Pilet, V. Lepetit, and P. Fua. Retexturing in the presence of complex illumination and occlusions. In *Proceedings of the International Symposium on Mixed and Augmented Reality*, 2007.
- J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2) :109–122, February 2008.
- D. Pizarro and A. Bartoli. Shadow resistant direct image registration. In *Proceedings of the Scandinavian Conference on Image Analysis*, 2007.
- D. Pizarro, J. Peyras, and A. Bartoli. Light-invariant fitting of active appearance models. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.
- J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images : a survey. *IEEE Transactions on Medical Imaging*, 22(8) :986–1004, August 2003.
- M. Prasad, A. Zisserman, and A. Fitzgibbon. Single view reconstruction of curved surfaces. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.
- S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proceedings of the International Conference on Computer Vision*, 2003.
- E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Proceedings of the International Conference on Computer Vision*, 2005.
- D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes. Nonrigid registration using free-form deformations : Application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8) :712–721, June 1999.
- D. Ruprecht and H. Muller. Free form deformation with scattered data interpolation methods. In *Geometric Modelling (Computing Suppl. 8)*, pages 267–281, 1993.
- M. Salzmann, S. Ilic, and P. Fua. Physically valid shape parameterization for monocular 3D deformable surface tracking. In *Proceedings of the British Machine Vision Conference*, 2005.
- M. Salzmann, R. Hartley, and P. Fua. Convex optimization for deformable surface 3D tracking. In *Proceedings of the International Conference on Computer Vision*, 2007a.
- M. Salzmann, V. Lepetit, and P. Fua. Deformable surface tracking ambiguities. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007b.
- M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2008.
- V. Scholz and M. Magnor. Cloth motion from optical flow. In *Proceedings of the Workshop on Vision Modelling and Visualization*, 2004.
- V. Scholz, T. Stich, M. Magnor, M. Keckeisen, and M. Wacker. Garment motion capture using color-coded patterns. In *Proceedings of SIGGRAPH*, 2005.

- S. Sclaroff and J. Isidoro. Active blobs. In *Proceedings of the International Conference on Computer Vision*, 1998.
- T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH*, 1986.
- C. Shannon. *A mathematical theory of communication*. Bell System, 1948.
- J. Shi and C. Tomasi. Good features to track. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1994.
- G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.
- P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *Proceedings of the European Conference on Computer Vision*, 1996.
- P. Taddei and A. Bartoli. Template-based paper reconstruction from a single image is well posed when the rullings are parallel. In *Proceedings of the NORDIA Workshop on Non-Rigid Shape Analysis and Deformation Image Alignment at CVPR*, 2008.
- J-P. Tardif, A. Bartoli, M. Trudeau, N. Guilbert, and S. Roy. Algorithms for batch matrix factorization with application to structure-from-motion. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2007.
- C. Tomasi and T. Kanade. Shape and motion from image streams : a factorization method. *International Journal of Computer Vision*, 9(2) :137–154, 1992.
- P. H. S. Torr and A. Zisserman. Feature based methods for structure and motion estimation. In *Proceedings of the Workshop on Vision Algorithms : Theory and Practice*, 1999.
- L. Torresani, D. Tang, G. Alexander, and C. Bregler. Tracking and modeling objects with rank constraints. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2001.
- L. Torresani, A. Hertzmann, and C. Bregler. Structure-from-motion : Estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5) : 878–892, May 2008.
- B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15 (8) :617–625, September 1997.
- A. Vedaldi and S. Soatto. Local features, all grown up. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2006.
- R. Vidal and D. Abretske. Nonrigid shape and motion from multiple perspective views. In *Proceedings of the European Conference on Computer Vision*, 2006.
- J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Proceedings of the International Conference on Computer Vision*, 2005.



- 
- J. Xiao, J. X. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion recovery. In *Proceedings of the European Conference on Computer Vision*, 2004.
- M. Yamamoto, P. Boulanger, J. A. Beraldi, M. Rioux, and J. Domey. Direct estimation of deformable motion parameters from range image sequence. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 1990.
- A. J. Yezzi and S. Soatto. Deformation : Deforming motion, shape average and the joint registration and approximation of structures in images. *International Journal of Computer Vision*, 53(2) :153–167, March 2003.