Approches orientées modèle pour la capture des mouvements du visage en vision par ordinateur

Marius Malciu

UNIVERSITÉ RENÉ DESCARTES – PARIS V
Centre Universitaire des Saints-Pères
UFR DE MATHÉMATIQUES ET INFORMATIQUE

Thèse présentée en vue de l'obtention du grade de Docteur de l'Université RENÉ DESCARTES – PARIS V

Discipline : Sciences de la Vie et de la Matière
Spécialité : Mathématiques et Informatique

Par

Marius MALCIU

Sujet de la thèse :

Approches orientées modèle pour la capture des mouvements du visage en vision par ordinateur

Soutenue le 20 décembre 2001, devant le jury composé de :

Monsieur	Georges STAMON	Président
Monsieur	Jean-Marie BECKER	Rapporteur
Monsieur	Michel DHOME	Rapporteur
Madame	Françoise PRÊTEUX	Directeur de thèse
Monsieur	Vasile BUZULOIU	Examinateur
Madame	Christine GRAFFIGNE	Examinateur

Remerciements

Débutés il y a plus de trois ans, les travaux présentés dans ce mémoire ont été effectués au sein de l'Unité de Projets ARTEMIS de l'Institut National des Télécommunications.

Je tiens tout d'abord à exprimer mes remerciements à :

Madame le Professeur Françoise Prêteux, Responsable de l'Unité de Projets ARTEMIS et directeur de cette thèse. Qu'elle soit assurée de ma sincère reconnaissance pour la qualité scientifique de la formation dont elle m'a fait bénéficier, pour l'infinie patience avec laquelle elle a dirigé mes travaux pendant toutes ces années et pour la confiance qu'elle m'a accordée.

Monsieur Georges Stamon, Professeur à l'Université Paris V, qui me fait l'honneur de présider le jury de soutenance et à lui témoigner mon respect et ma gratitude.

Monsieur Jean-Marie Becker, Maître de Conférences à CPE Lyon, et Monsieur Michel Dhome, Directeur de Recherche du CNRS à l'Université Blaise Pascal Clermont-Ferrand, qui me font l'honneur de s'intéresser à ce travail, et qui ont accepté d'en être les rapporteurs diligents.

Madame Christine Graffigne, Professeur à l'Université Paris V, qui me fait l'honneur de participer à ce jury. Qu'elle soit assurée de mon respect et de ma sympathie.

Monsieur Vasile Buzuloiu, Professeur à l'Université POLITEHNICA de Bucarest auquel je dois ma formation en traitement d'images et mes premiers pas en recherche. Que ce Professeur exemplaire, scientifique passionné et, avant tout, homme de cœur, soit assuré de ma profonde reconnaissance.

Mes remerciements vont également à Nicolas Rougon, Maître de Conférences à l'INT, pour nos discussions scientifiques et amicales et à Mesdames Nicole Teste et Evelyne Taroni, pour leur bonne humeur au quotidien et pour leur aide précieux dans les arcanes administratives.

Que Catalin Fetita, Maître de Conférences à l'INT, trouve ici l'expression des mes sincères remerciements pour son soutien indéfectible notamment quand il fallait respecter certaines contraintes temporelles ...

Je tiens également à remercier tout aussi chaleureusement Marius Preda et Titus Zaharia, doctorants à l'Unité de Projets ARTEMIS qui, par leur aide et leur bonne humeur, m'ont soutenu et encouragé pendant toutes ces années.

Enfin, je souhaiterais remercier mes amis et collègues qui ont contribué, à divers titres, au bon déroulement de cette thèse : Maria Banu, Răzvan Beuran, Mihai Ciuc, Mircea Curilă, Sorin Curilă, Cristina Dan, Ionuţ Deaconeasa, Anca Flueraşu, Melania Ionescu, Mihai Ivanovici, Carlos Martin, Mihai Mitrea, Gérard Mozelle, Ilina et Tudor Murgan, Laura Necula, Louis-Thomas Nessi, Caroline Petitjean, Augustin Radu, Gabriela et Vlad Valica, Constantin Vertan.

A ma famille

Table des matières

1 Suivi des mouvements du visage : état de l'art **11**

1.1 Principe du suivi 3D du visage à base de modèle 11

1.1.1 Modèles de visage 11

1.1.2 Définition et mise en correspondance de primitives 14

1.1.2.1 Primitives d'images 14

1.1.2.2 Primitives de modèle 15

1.1.2.3 Mise en correspondance des primitives d'images et de modèle . . 15

1.2 Suivi des mouvements globaux du visage 16

1.2.1 Méthodes globales 17

1.2.1.1 Méthodes à base de flot optique 17

1.2.1.2 Méthodes à base de texture 20

1.2.2 Méthodes locales 22

1.3 Suivi des mouvements locaux du visage 23

1.3.1 Suivi de points caractéristiques 24

1.3.1.1 Suivi non-contraint 25

1.3.1.2 Suivi de points reliés 25

1.3.2 Suivi de régions d'intérêt 26

1.3.2.1 Suivi sans modèle géométrique 26

1.3.2.2 Modèles rigides 27

1.3.2.3 Modèles déformables 27

1.3.2.3.a Lignes et courbes 27
1.3.2.3.b Contours actifs 28
1.3.2.3.c Prototypes déformables 30
1.3.2.3.d *Active shape models* 30
1.4 Les approches adoptées et développées 32

2 Estimation de la pose 3D du visage par modèle d'objet 35

2.1 Positionnement du problème 35
2.2 Techniques de base 37
2.2.1 Estimation du mouvement apparent de la scène : flot optique 37
2.2.1.1 Généralités 37
2.2.1.2 Méthodes de calcul 38
2.2.1.2.a Méthodes différentielles : l'algorithme de Horn et Schunk 39
2.2.1.2.b Méthodes par corrélation : les algorithmes de Lucas et Kanade et de Quénot 41
2.2.1.3 Approches multirésolution 44
2.2.1.4 Implantation algorithmique 48
2.2.1.5 Résultats : analyse qualitative et quantitative 49
2.2.2 La méthode d'optimisation du simplexe 51
2.3 Modélisation par flot coloré 55
2.3.1 Modélisation 3D de la tête 56
2.3.2 Primitives 2D de l'image et 3D du modèle 58
2.3.3 Principe de la méthode de recalage 3D/2D 60
2.3.4 Estimation en présence de grande translation : compensation du mouvement translationnel dominant 61
2.3.5 Résultats 63
2.4 Conclusion 67

3 Estimation robuste de la pose 3D du visage 71

3.1 Avant-propos 71
3.2 Interpolation temporelle : approche physique par groupe de paquets d'onde . . . 72
3.2.1 Interpolation linéaire 72
3.2.2 Analogie ondulatoire : le modèle du front d'onde 73
3.2.3 Modélisation par groupe de paquets d'onde 75
3.3 Estimation robuste à base d'indice de visibilité 77
3.4 Analyse des occultations 79
3.4.1 Positionnement du problème 79
3.4.2 L'algorithme des *k-moyennes* 80
3.4.3 Classification au sens du mouvement 81
3.4.4 Analyse de similarité de mouvements 83
3.5 Méthode robuste d'estimation de la pose 3D et résultats 86
3.6 Conclusion 88

4 Suivi de primitives de visage 93

4.1 Positionnement du problème 93
4.2 Principe du recalage par prototypes déformables et outils associés 94
4.2.1 Prototypes déformables 94
4.2.2 Modélisation géométrique à base de splines 96
4.2.2.1 Généralités 97
4.2.2.2 B-splines 98
4.2.3 Segmentation floue sous contrainte spatiale 101
4.2.3.1 Modèle de mélange 102
4.2.3.2 Estimation des paramètres du mélange 104
4.2.3.2.a Maximum de vraisemblance 104
4.2.3.2.b L'algorithme EM 104
4.2.3.2.c Analogie avec la classification floue. Régularisation spatiale 108

4.3 Prototypes déformables compatibles MPEG-4 109
4.3.1 Descripteur de visage MPEG-4 110
4.3.1.1 Paramètres de description du visage (FDPs) 110
4.3.1.2 Paramètres d'animation du visage (FAPs) 110
4.3.2 Construction de prototypes déformables compatibles MPEG-4 115
4.3.2.1 Modélisation géométrique 115
4.3.2.2 Contraintes internes 119
4.3.2.3 Détection de la configuration de l'œil 122
4.4 L'interaction prototype - image 123
4.4.1 Définition et extraction des primitives d'image 124
4.4.2 Contraintes externes 125
4.5 Suivi de la bouche et des yeux 130
4.5.1 Algorithme de suivi 130
4.5.2 Résultats 131
4.6 Application à l'animation MPEG-4 d'avatar 132
4.7 Conclusion 134

Conclusion et perspectives **139**

Liste de publications associées **143**

Bibliographie **144**

Liste des figures

1.1 Le modèle *Candide*, dans ses trois versions les plus courantes. 13

1.2 Modèles paramétriques de tête. 14

1.3 Principe de *feedback* dans une approche coopérative analyse/synthèse. 19

2.1 Calcul du chemin optimal de recalage dans l'algorithme de Quénot. 43

2.2 Pyramide d'image à 4 niveaux. 45

2.3 Estimation du champ de vitesses par l'algorithme de Horn et Schunk en version multirésolution. 47

2.4 Différentes estimations du flot optique. 50

2.5 Fonctions de répartition des erreurs d'estimation du champ de déplacements correspondants aux algorithmes de Horn et Schunk, de Lucas et Kanade et de Quénot. 52

2.6 Les opérations géométriques de base de l'algorithme du simplex. 54

2.7 Ajustement des points de contour de la tête par une série trigonométrique. 58

2.8 Surfaces synthétisées correspondant aux différents ordres de la série trigonométrique. 59

2.9 Principe du recalage 3D/2D à base d'un modèle géométrique. 60

2.10 Le principe de compensation du mouvement translationnel dominant. 62

2.11 Recalage 3D à base de flot coloré, avant et après la compensation du mouvement translationnel dominant. 64

2.12 Images provenant des séquences de test. 65

2.13 Fonctions de distribution des erreurs d'estimation pour les six paramètres de la pose 3D. 66

2.14 La méthode par flot coloré : suivi 3D effectué avec succès. 67

2.15 La méthode par flot coloré : échec du suivi 3D. 68

3.1 Schéma de l'interpolation simple du flot. 73

3.2 Interpolation spatio-temporelle linéaire. 74

3.3 Front d'onde localisé à l'origine. 76

3.4 Interpolation par groupe de paquets d'onde. 78

3.5 Représentation de l'indice de visibilité du modèle dans la position de référence. . 79

3.6 Résultat de l'algorithme des *k-moyennes* appliqué directement aux vecteurs de déplacement. 82

3.7 Classification au sens du mouvement en présence d'occultations du visage. 84

3.8 Classification au sens du mouvement pour un visage parlant. 85

3.9 Schéma synoptique de la méthode robuste d'estimation de la pose 3D. 87

3.10 Fonctions de distributions des erreurs d'estimation robuste pour les six paramètres de la pose 3D. 89

3.11 Résultats obtenus par la méthode robuste pour des rotations et pour des occultations. 90

3.12 La méthode robuste par flot coloré : suivi 3D effectué avec succès. 91

4.1 Exemple de B-splines cubiques. 100

4.2 Les paramètres FDPs regroupant la description de forme par maillage 3D et l'information de texture. 111

4.3 Points caractéristiques des FDPs tels que définis dans la norme MPEG-4. 112

4.4 Caractéristiques faciales de référence pour la définition des *Facial Animation Parameter Units* (FAPU). 114

4.5 Modèle géométrique du prototype de la bouche. 115

4.6 Modèles géométriques des prototypes de l'œil ouvert et de l'œil fermé. 116

4.7 Succès de la modélisation des contours des lèvres par des arcs de paraboles. . . . 117

4.8 Echec de la modélisation des contours des lèvres par des arcs de paraboles. . . . 118

4.9 Interpolation d'un ensemble de points du plan par une courbe spline. 119

4.10 Modélisation de la bouche et des yeux par des B-splines cubiques reliant les points caractéristiques MPEG-4. 120

4.11 Graphes associés aux points caractéristiques de la bouche et de l'œil ouvert, spécifiant l'interdépendance des déformations. 121

4.12 Symétries locales exprimées sur un contour de lèvre et d'œil. 121

4.13 Classification floue sous contrainte spatiale pour la détection de la configuration de l'œil. 123

4.14 Classification obtenue par l'algorithme NEM appliqué pour trois classes initialisées de manière arbitraire. 124

4.15 Segmentation floue sous contrainte spatiale, dans la région de la bouche. 126

4.16 Segmentation floue sous contrainte spatiale, dans la région de la bouche (suite). . 127

4.17 Segmentation floue sous contrainte spatiale, dans la région de la bouche (suite). . 128

4.18 Segmentation floue sous contrainte spatiale, dans la région de la bouche (suite). . 129

4.19 Exemples de prototypes recalés sur les images correspondant à deux séquences de test. 133

4.20 Transformation des données primaires en FAPs à transmettre à un codeur générant un flux MPEG-4. 135

4.21 Animation MPEG-4 d'avatar à partir d'une séquence naturelle. 136

4.22 Animation MPEG-4 d'avatar à partir d'une séquence naturelle (suite). 137

Liste des tableaux

1.1 Primitives d'image utilisables pour le suivi des mouvements locaux du visage. . . 24

2.1 Valeurs moyennes et écarts types des erreurs d'estimation du champ de déplacements par différents algorithmes. 51

4.1 Les visèmes définis dans la norme MPEG-4. 111

4.2 Les expressions faciales définies dans la norme MPEG-4 et leur description textuelle. 113

4.3 *Facial Animation Parameter Units* (FAPU) définies en fonction des distances entre des points caractéristiques du visage. 114

4.4 Les pondérations des différentes composantes énergétiques des prototypes. 132

Chapitre 1

Suivi des mouvements du visage : état de l'art

Après un bref rappel du principe du suivi 3D du visage à base de modèle, ce chapitre décrit les méthodes génériques de la littérature concernant le suivi du visage, aussi bien dans sa globalité que dans ses déformations locales.

1.1 Principe du suivi 3D du visage à base de modèle

Dans un cadre très général et non-formalisé, le suivi 3D du visage à base de modèle dans des séquences vidéos consiste en une procédure de mise en correspondance entre deux ensembles de primitives, le premier associé aux images 2D de la séquence et l'autre correspondant au modèle 3D. Conformément à cette formulation, un tel algorithme de suivi implique :

- une modélisation 3D du visage, c'est-à-dire la sélection ou la construction d'un modèle de visage ;
- la définition et l'extraction de primitives de l'image et du modèle ;
- la mise en correspondance effective de ces primitives.

Chacune de ces étapes comporte différents choix méthodologiques ou algorithmiques que nous allons présenter et analyser par la suite.

1.1.1 Modèles de visage

Les modèles de visage utilisés pour le suivi de déformations et l'estimation de la pose 3D ont connu une évolution rapide en termes de complexité et de qualité de représentation. La grande

variété de modèles aujourd'hui disponibles au sein de la communauté scientifique peut être structurée en modèles régionaux, analytiques ou paramétriques.

Les modèles régionaux, les plus élémentaires de toutes les modélisations envisageables, ne prennent en compte que quelques points remarquables du visage comme par exemple, les coins des yeux et les commissures des lèvres [Gee94, Stiefelhagen98]. Contenant très peu d'information sur la géométrie du visage, ils sont conçus principalement pour estimer la pose 3D de la tête, et sont incapables de capturer les déformations faciales.

Les modèles analytiques quant à eux permettent une représentation compacte et une manipulation facile au moins formellement, ce qui fait leur succès dans les techniques de suivi 3D où ils sont souvent utilisés sous forme de surfaces planaires [Black95], cylindriques [LaCascia98], ellipsoïdales [Basu96, Grammalidis00], ou de super-quadriques [Zhang.Y00].

Une autre classe largement utilisée regroupe les modèles paramétriques offrant une description formelle des déformations du visage. Le modèle 3D paramétré de visage le plus connu est *Candide*. Il a été développé dans les années 80 par Rydfalk [Rydfalk87] à l'Université de Linköping en Suède pour des applications de codage et d'animation de visage à base de modèle. Il s'agit d'un maillage 3D triangulaire muni d'un certain nombre de paramètres de déformation nommés *action units*, dérivés de l'étude de Ekman et Friesen [Ekman77] sur la physiologie des expressions faciales. Adapté plutôt aux moyens de calcul de l'époque, il reste très simple, se composant de 75 sommets et 100 triangles. Peu utilisé dans sa forme originelle, ce modèle a subi des évolutions successives donnant naissance à trois versions (Figure 1.1). La première et la plus répandue, *Candide*-1, contient 79 sommets, 108 triangles et 11 *action units* (Figure 1.1(a)). La deuxième version, *Candide*-2 [Welsh91], incluant également les cheveux et les dents, comporte 160 sommets, 238 triangles, mais seulement 6 *action units* (Figure 1.1(b)). La troisième version [Ahlberg01], dérivée directement du modèle original, a eu pour objectif de simplifier l'animation à travers les *Facial Animation Parameters* définis dans la norme MPEG-4 [MPEG-4]. Pour cela, environ 20 sommets ont été ajoutés, la plupart d'entre eux correspondant aux points spécifiques de MPEG-4 (Figure 1.1(c)).

Toutefois, les modèles *Candide*, étant assez simples, sont limités au niveau de la qualité du rendu, comparés aux modèles maillés d'aujourd'hui, contenant parfois plusieurs centaines de sommets, qui permettent une représentation beaucoup plus détaillée du visage (Figure 1.2(a)).

Un autre type de modèles paramétriques-analytiques sont les modèles à base de B-splines [Eisert98] ayant comme avantage le lissage 3D des surfaces rendues, conduisent à des représentations réalistes du visage. Dans ce cas, les déformations du visage sont paramétrisées par les points de contrôle des B-splines. La Figure 1.2 (b) montre un exemple de modèle 3D de tête

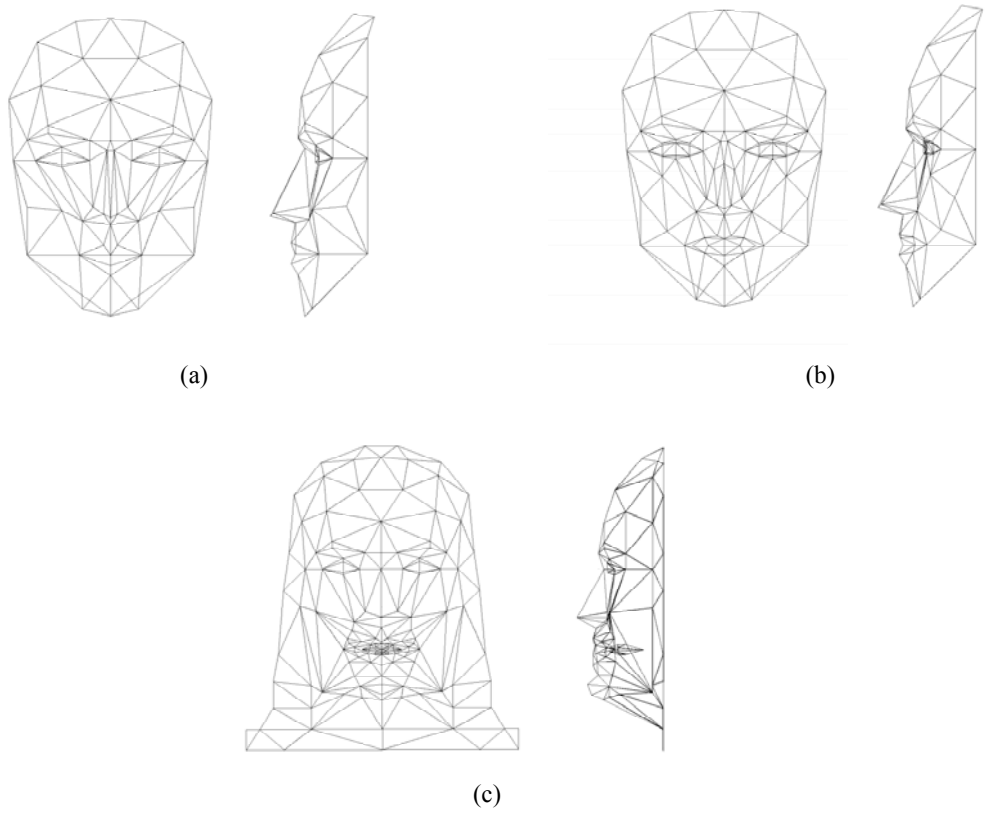
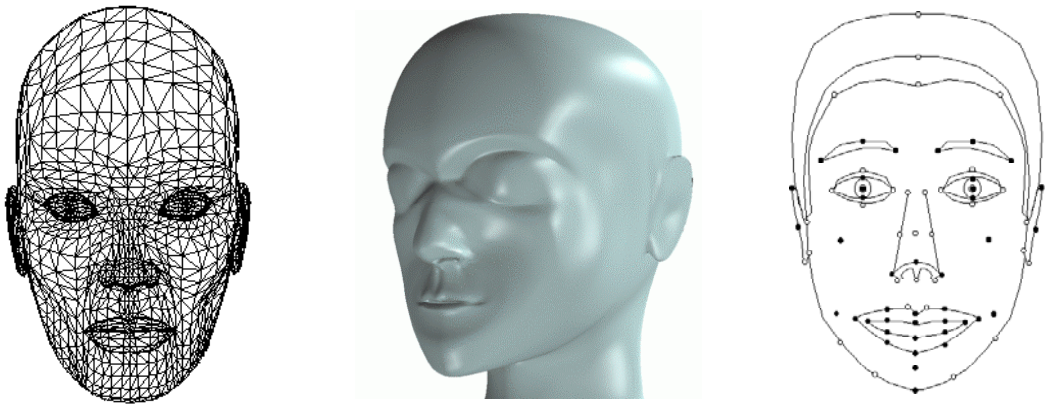


Figure 1.1 : Le modèle *Candide*, dans ses trois versions les plus courantes. (a) Candide-1, (b) Candide-2, (c) Candide-3.

réalisé avec des B-splines.

Dans sa première version de 1999, le standard MPEG-4 (*Moving Picture Expert Group*) normalise déjà le visage ainsi que ses déformations en termes de *Face Definition Parameters* (FDPs) et de *Face Animation Parameters* (FAPs) (Figure 1.2(c)). Les FDPs sont des points du visage qui décrivent d'une manière compacte sa géométrie, tandis que les FAPs paramétrisent les déplacements que les FDPs peuvent subir. La norme n'impose toutefois pas un modèle 3D spécifique, les utilisateurs ayant ainsi la liberté d'animer leur propre modèle.

Des modèles paramétriques encore plus sophistiqués qui s'inspirent de l'anatomie humaine, simulant la peau et les tissus en terme d'actions musculaires ont été développés [Terzopoulos93], mais ceux-ci sont plutôt appliqués dans le contexte de la synthèse d'avatars, qui dépasse le cadre de notre étude.



(a) (b) (c)

Figure 1.2 : Modèles paramétriques de tête : (a) modèle maillé complexe, (b) modèle à base de B-splines, (c) description MPEG-4 du visage à l'aide des *Face Definition Parameters* (FAPs).

1.1.2 Définition et mise en correspondance de primitives

Comme précisé précédemment, réaliser le suivi 3D du visage à base de modèle dans des séquences vidéos revient à apparier deux ensembles de primitives, l'un associé aux images 2D de la séquence et l'autre au modèle 3D utilisé. Nous allons présenter par la suite les différents choix possibles pour ces ensembles de primitives et la manière de réaliser leur mise en correspondance.

1.1.2.1 Primitives d'images

Dans notre contexte, les primitives d'image représentent un ensemble d'informations défini pour caractériser de manière spécifique la région du visage et l'individualiser par rapport à d'autres objets présents dans la scène. Dans l'hypothèse ou l'arrière plan ne subit pas de déplacements importants, l'information de mouvement 2D peut être utilisée avec succès pour estimer la pose 3D de la tête [Black95, Basu96], les déformations non-rigides du visage [Li93, Choi94, Bozdagi94], ou même la géométrie 3D de la tête [Azarbayejani95]. Cette information de mouvement est exprimée en termes de champ de déplacement 2D (flot optique) défini sur le support de l'image. L'hypothèse sous-jacente au calcul du flot optique renvoie à la conservation de la luminance du pixel entre deux images successives de la séquence [Horn81]. Notons par $(I_n)_n$ la séquence vidéo et par (x, y) les cordonnées spatiales d'un pixel dans le plan de l'image. La contrainte précédemment énoncée s'exprime comme suit :

$$I_n\left(x - \Delta x_n, y - \Delta y_n\right) = I_{n+1}\left(x, y\right) \ . \tag{1.1}$$

De plus, postulant des petits déplacements, l'équation précédente est approchée par :

$$\frac{\partial I_n}{\partial x}\,\Delta x_n + \frac{\partial I_n}{\partial y}\,\Delta y_n + (I_{n+1} - I_n) = 0, \tag{1.2}$$

et est connue sous le nom d'équation du flot optique. Précisons que cette équation n'a pas de solution unique. Pour calculer le champ de déplacement, il est nécessaire d'imposer des hypothèses complémentaires. Afin de surmonter la contrainte de petits déplacements, des représentations multi-échelle (ou multirésolution) sont possibles [Enkelmann86, Glazer87]. Celles-ci seront reprises et détaillées au chapitre 2.

D'autres représentations exploitent directement la texture du visage pour le suivi 3D rigide [LaCascia98] ou déformable [Colmenarez97]. En outre, les contours de l'image peuvent servir pour extraire la silhouette du visage [Reinders95], ou pour localiser des points caractéristiques [Stiefelhagen98]. Diverses représentations et modélisations de la teinte de chair permettent une localisation grossière du visage [Sobottka96].

1.1.2.2 Primitives de modèle

Les primitives de modèle renvoient à la géométrie ou la topologie 3D du modèle utilisé pour l'étape de suivi et sont spécifiques de celui-ci. Reprenant la classification de la section 1.1.1, dans le cas des modèles régionaux, les primitives associées sont pratiquement confondues avec le modèle en raison de sa simplicité et sont représentées par un ensemble de coordonnées spatiales définissant, par exemple, les coins des yeux ou les commissures des lèvres.

Dans le cas des modèles analytiques, les primitives considérées sont les paramètres individualisant l'équation de la surface 3D du modèle. Pour les modèles paramétriques, la géométrie globale et les déformations locales sont exprimées à travers un jeu de paramètres qui constitue également l'ensemble de primitives associées au modèle. Par exemple, ces primitives sont définies sous forme de *action units* pour les modèles *Candide*, de nœuds contrôlant les modèles à base de B-splines, de *FAPs* dans le cadre des modèles MPEG-4, etc.

1.1.2.3 Mise en correspondance des primitives d'images et de modèle

La mise en correspondance entre les primitives d'image et celles du modèle consiste en une procédure d'optimisation dans un certain espace de paramètres caractérisant la configuration du modèle. Pour cela, deux options s'offrent :

- considérer un problème typique d'estimation ;

- interpréter la mise à jour des paramètres du modèle le long de la séquence, en termes de prédiction.

Chacune de ces deux approches présente des avantages spécifiques ainsi que des inconvénients que nous allons discuter.

Dans le premier cas, les paramètres du modèle résultent de la minimisation d'un estimateur défini en terme d'écart entre les deux ensembles de primitives. Choisir un estimateur caractérisé par un nombre réduit de minima locaux et mettre en œuvre un algorithme de minimisation performant peut conduire à des résultats extrêmement robustes. Précisons qu'en général les algorithmes robustes d'optimisation sont coûteux en volume de calcul. Parmi les méthodes d'optimisation les plus utilisées, nous trouvons la descente de gradient classique [Black95], ou sous la forme de Levenberg-Marquardt [LaCascia98], les réseaux neuronaux [Fukuhara93], les algorithmes de relaxation stochastique [Bozdagi94, Viola95] et la descente du simplexe [Basu96].

En ce qui concerne la deuxième option, la prédiction offre une formule permettant le rafraîchissement de l'état du modèle dès qu'une nouvelle image arrive. Les techniques prédictives nécessitent une étape laborieuse de modélisation aussi bien pour les données utiles que pour les perturbations. En revanche, les algorithmes sont rapides, permettant parfois des implantations en temps réel [Strom99]. Connu comme une technique robuste et puissante, le filtrage de Kalman sous ses différentes formulations est le plus fréquemment utilisé [Azarbayejani95, Colmenarez97].

Alors, que choisir entre estimation et prédiction ? Seul le contexte de l'application envisagée est en mesure d'influer sur la réponse. Qu'en est-il dans le cas du suivi des mouvements du visage ?

1.2 Suivi des mouvements globaux du visage

Cette section traite des différentes méthodes utilisées pour le suivi du visage. Précisons tout d'abord que nous nous intéressons uniquement aux techniques de suivi 3D adaptées aux séquences vidéos monoscopiques. Dans une classification très générale, les différentes approches existantes peuvent être divisées en deux grandes catégories, selon la manière d'interpréter et d'utiliser l'information contenue dans la région du visage. Une première catégorie, recouvrant les *méthodes globales*, traite de manière unitaire tous les pixels du visage en les combinant au sein de différents modèles de mouvement et de schémas d'optimisation. La seconde classe, correspondant aux *méthodes locales*, considère que l'information pertinente du visage est localisée dans certaines régions de l'image, en général correspondant aux éléments faciaux. Ces méthodes, étant focalisées sur les particularités du visage et mettant l'accent sur l'extraction des primi-

tives d'image et sur leur interprétation en terme de candidats aux éléments faciaux recherchés, sont en général plus restrictives que les méthodes de la première catégorie par rapport à la configuration de la scène et aux conditions d'éclairement.

1.2.1 Méthodes globales

La démarche la plus souvent utilisée, dans ce contexte, consiste à définir un modèle 3D de visage muni d'un modèle de mouvement. Dans ce cas, l'estimation de la pose 3D du visage s'effectue par une procédure d'appariement entre le modèle du visage projeté et les primitives d'image, qu'il s'agisse de flot optique ou de texture. Le choix des modèles et des primitives d'image différencie entre les approches existant dans la littérature.

1.2.1.1 Méthodes à base de flot optique

La méthode de suivi 3D proposée par Black et Yacoob [Black95] repose sur une approximation planaire du visage couplée au modèle de projection perspective. Le champ de déplacement projeté, généré par le mouvement 3D du plan d'approximation, est modélisé par des polynômes du second degré. Le modèle polynomial, reporté dans l'équation du flot optique, permet de définir un estimateur de la pose 3D. Les paramètres du modèle, notamment les coefficients polynomiaux, sont calculés dans le cadre d'un schéma d'estimation robuste décrit dans [Black93]. En effet, introduisant dans l'expression de l'estimateur une norme robuste munie d'un paramètre de contrôle, plusieurs étapes de minimisation par descente de gradient sont effectuées en faisant varier ce paramètre, ce qui permet de réduire progressivement l'influence des *outliers* et ainsi de diminuer les erreurs d'estimation. La même technique, appliquée sur de petites régions d'intérêt contenant la bouche, les yeux et les sourcils et prenant en compte des modèles de mouvement capables de gérer les déformations de ces éléments faciaux, conduit à une caractérisation quantitative de l'expression faciale. Celle-ci est utilisée ensuite dans une procédure de reconnaissance temporelle, en s'appuyant sur un ensemble de six expressions, considérées fondamentales par les auteurs. Même si pour une telle application de reconnaissance, le modèle planaire de visage peut être utilisé avec un certain succès (reconnaissance correcte dans 90% des cas), il est trop simple pour pouvoir espérer un suivi 3D précis et robuste sur des séquences étendues, d'autant plus que l'algorithme ne prend pas en compte les discontinuités du champ de déplacement introduites par les éventuelles occultations partiales du visage.

Utilisant un modèle plus adapté à la géométrie du visage, notamment un ellipsoïde, l'approche proposée par Basu *et al.* [Basu96], diffère de l'algorithme précédent par la manière d'utiliser l'information de mouvement. Au lieu d'intégrer le mouvement du modèle dans la contrainte

de conservation de la luminance, l'estimateur de la pose 3D est défini en termes d'écart entre un champ de déplacement théorique, résultant du mouvement rigide du modèle projeté dans le plan de l'image, et un champ de déplacement expérimental, calculé entre deux images successives de la séquence, à base d'un algorithme de flot optique. La projection est effectuée conformément au modèle perspectif. Les paramètres de la pose 3D du modèle résultent de la minimisation de cet estimateur au moyen de l'algorithme du simplexe. Les auteurs ne fournissent pas de détails concernant la précision de l'algorithme de flot optique utilisé et n'indiquent pas les limitations par rapport à l'amplitude du mouvement. Dans une étape de validation, réalisée sur une séquence de synthèse (≈150 images), l'estimation reste stable et dans une limite d'erreur de 5 pixels pour les translations et de 10° pour les rotations. Toutefois, utiliser de manière brute le flot optique ne permet pas de gérer les occultations du visage. Le fait que le flot optique puisse apporter de l'information sur les occultations est traité plus récemment par Zhang et Kambhamettu [Zhang.Y00]. En reprenant la méthode de Basu *et al.*, pour un modèle de tête de type super-quadrique, l'algorithme proposé s'appuie sur une élimination successive des pixels du visage susceptibles d'être occultés, selon un critère de seuil, pendant plusieurs étapes d'optimisation. Cependant, cette élimination, effectuée de manière intuitive et sans une formulation mathématique rigoureuse, risque de conduire à des faux appariements.

Li *et al.* [Li93] s'inscrivent dans un contexte de codage vidéo à base de modèle 3D, et utilisent l'équation de mouvement de Helmholtz, posant que le mouvement ponctuel d'un objet déformable résulte de la superposition d'un déplacement global rigide et d'un déplacement local exprimé comme une fonctionnelle linéaire. Utiliser directement ce modèle, sans aucune hypothèse supplémentaire, conduit à un problème mal posé. Pour réduire la dimension de l'espace des solutions, les auteurs utilisent la paramétrisation du modèle *Candide* en terme de *action units* et définissent un modèle mathématique de mouvement qui regroupe hypothèse de Helmholtz, projection perspective et contrainte de flot optique. En conséquence, il en découle un système linéaire d'équations ayant comme inconnues les paramètres globaux de la pose et les paramètres locaux de déformation du modèle. Le système étant surdéterminé, il est résolu au sens des moindres carrés. Le résultat, ainsi qu'un placage de texture sont utilisés pour synthétiser la séquence originelle. En outre, l'algorithme est muni d'un module de prédiction linéaire pour prendre en compte des grands mouvements entre les images successives et d'un module de correction de mouvement, pour diminuer l'accumulation des erreurs. Ce dernier, fondé sur le principe du *feedback*, ré-applique l'algorithme d'estimation des paramètres entre chaque image synthétisée et l'image originelle correspondante, ce qui permet de calculer des correcteurs d'erreurs (Figure 1.3). Les performances de l'algorithme appliqué à des séquences de synthèse sont exprimées en terme d'erreur absolue d'estimation des paramètres de la pose 3D et des *action units*. Nous considérons que l'utilisation directe d'un modèle maillé pour estimer les mouve-

ments du visage pose des problèmes d'initialisation au début de la séquence. Malheureusement, cet aspect n'est pas explicité par les auteurs.

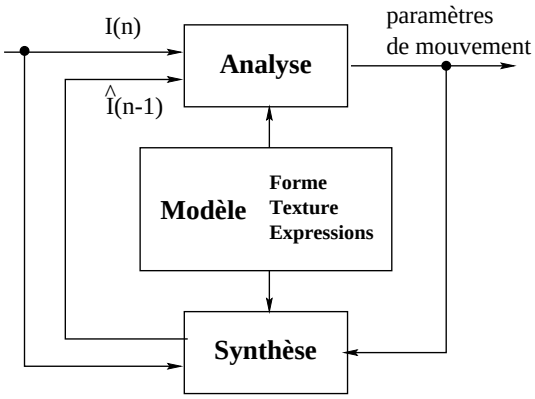


Figure 1.3 : Principe de *feedback* dans une approche coopérative analyse/synthèse.

Dans une démarche similaire, y compris l'utilisation de la boucle de *feedback*, Eisert et Girod [Eisert98] estiment les mouvements rigides et non-rigides du visage par l'intermédiaire des FAPs définis par la norme MPEG-4. Le maillage 3D utilisé est généré sur une surface modélisée par des B-splines dont les points de contrôle sont positionnés en concordance avec les FDPs. Disposant de technique de scannérisation 3D, la calibration et la texturation du modèle sont effectuées de manière précise. L'avantage de cette technique est la qualité de l'image de synthèse rendue.

Dans le même cadre du codage à base de modèle, Choi *et al.* [Choi94] proposent une méthodologie modulaire comportant deux étapes de complexité croissante : une estimation de la pose 3D de la tête (adaptation globale de la pose du modèle) suivie d'une estimation des déformations du visage (adaptation locale de la forme du modèle). Dans la première étape, l'équation du flot optique, appliquée à chaque pixel de la projection orthographique d'un modèle 3D maillé, considéré comme rigide, conduit à un système linéaire sur-déterminé. La solution, calculée au sens des moindres carrés, représente l'estimation de la pose 3D de la tête, qui sert à initialiser le modèle pour la seconde étape. Le même principe, appliqué au modèle paramétré en terme de *action units*, permet d'estimer les déformations du visage. La séquence est synthétisée par un placage de texture conduisant à un débit de transmission estimé à 10.5 Kbits/s, pour le modèle considéré. L'initialisation du modèle sur la première image de la séquence s'effectue par une mise en correspondance entre les contours extraits du visage et les arêtes du maillage, mais les détails de la procédure ne sont pas précisés. En se prêtant à des améliorations à chaque niveau, l'idée du traitement modulaire a été reprise et développée ultérieurement dans [Reinders95, Kampmann97, Zhang.L98].

Une approche plus élaborée, mettant en œuvre les aspects photométriques liés à la formation

de l'image est présentée dans [Bozdagi94]. Dans une première étape de traitement, la direction d'illumination de la scène est estimée, ce qui permet d'incorporer l'hypothèse lambertienne dans l'équation spatio-temporelle du flot optique. Cette démarche, étant d'ailleurs bien connue dans les techniques d'estimation de la structure et du mouvement dans une scène par flot optique [Verri89, Pentland91], n'a pas été utilisée auparavant dans le cas particulier du visage. Modélisant la tête par un maillage 3D triangulaire et considérant la projection orthographique, la contrainte de flot optique avec des effets photométriques est appliquée en chaque pixel du visage, conduisant à un système sur-détermine d'équations non linéaires dont les inconnues sont les cinq paramètres de la pose 3D (paramètres globaux), d'une part et les normales aux facettes du modèle (paramètres de structure, locaux), d'autre part. La solution de ce système, formulée au sens des moindres carrés et obtenue à travers un schéma de relaxation stochastique, représente l'estimation simultanée des paramètres globaux et locaux du modèle. Utilisant le modèle *Candide*, l'algorithme est testé dans un contexte d'analyse-synthèse pour la vidéotéléphonie, aussi bien sur des séquences synthétiques que sur des séquences réelles. Un placage de texture permet d'exprimer quantitativement les performances de la méthode en termes d'erreur quadratique moyenne entre la séquence originelle et celle synthétisée à 5 niveaux de gris près sur 256. Le problème d'initialisation du maillage au début de la séquence n'est pas détaillé.

Notons que les algorithmes utilisant sans précautions l'équation 1.2 du flot optique, qui représente en fait une approximation linéaire de la contrainte de conservation de luminance, ne sont pas capables de gérer les rotations de grande amplitude même si un schéma multirésolution est envisagé, étant donné que les rotations sont invariantes aux homothéties 2D.

1.2.1.2 Méthodes à base de texture

A l'encontre des approches précédentes reposant sur le flot optique, la démarche adoptée par LaCascia *et al.* [LaCascia98] consiste à utiliser directement la texture du visage. Le principe de l'algorithme revient à minimiser l'écart entre deux projections sur un modèle cylindrique, de la texture du visage correspondant à deux images successives de la séquence. Cela est équivalent à un recalage 2D/3D/2D en texture, par rapport aux paramètres de la pose 3D du modèle, dans un système de projection perspective. L'utilisation de la norme lorentzienne, issue de la théorie de l'estimation robuste, dans l'expression de la fonction objectif, permet de limiter l'influence des *outliers*. La minimisation est accomplie via un algorithme robuste de recalage par transformations projectives [Gleicher97] et capable de gérer des mouvements de grande magnitude. En outre, en s'inspirant de la modélisation de Black et Yacoob, les auteurs traitent aussi des mouvements non-rigides du visage. Malheureusement, les résultats numériques indiquant la précision de l'estimation ne sont pas fournis. Même si l'algorithme se rend robuste par rapport aux

mouvements rapides et aux occultations du visage, nous considérons que le modèle cylindrique et trop simpliste pour pouvoir capturer de manière exacte les mouvements du visage.

Plus génériques que les méthodes précédentes, les approches de type *structure from motion* (SfM) tentent de restituer simultanément les mouvements 3D, la focale de la caméra et le modèle 3D de l'objet à partir d'un ensemble de points de l'objet suivi dans la séquence vidéo. Ayant ses racines en photogrammétrie [1] [Slama80] et en vision par ordinateur avec les premiers résultats de reconstruction de scènes 3D à partir d'images stéréos [Marr76], la SfM propose des solutions alternatives dans un large domaine d'applications, telles que : reconstruction de modèles 3D [Debevec96], estimation de mouvement 3D [Azarbayejani95], calibration de la caméra [Blake92], codage 3D de séquences d'images [Strinzis99], robotique [Andreff01], etc.

Dans le cadre des applications de suivi d'objets 3D dans des séquences vidéos, la SfM utilise principalement deux formalismes : l'un linéaire, où les paramètres liés à la forme de l'objet et au mouvement de la caméra s'expriment en fonction des mesures 2D sur l'image à travers une matrice fondamentale et l'autre non-linéaire, à base de minimisation d'une fonction de coût ou de filtrage de Kalman. Toutefois, les premières approches se heurtent d'une part, à une instabilité en présence du bruit et de configurations dégénérées par rapport à la géométrie de projection [Tomasi92] et, d'autre part, à la complexité du traitement [Faugeras92].

Les approches non-linéaires, plus souples et plus précises, estiment la structure 3D de l'objet et les paramètres de mouvement soit par minimisation d'une fonction de coût selon la technique épipolaire simple (deux cadres et projection perspective) ou étendue à plusieurs cadres, soit par filtrage de Kalman utilisé aussi bien en version simpliste pour lisser le résultat d'estimation de la technique épipolaire [Oliensis91, Soatto93], que sous forme d'un véritable estimateur d'état à partir des mesures 2D, tout en tenant compte de la dynamique interne, complexe, de l'état [Broida90]. Dans un formalisme non-linéaire récursif, Jebara *et al.* [Jebara99] proposent un système dynamique capable de reconstituer simultanément la structure d'un objet 3D, le movement 3D et la géométrie interne de la caméra, à partir d'un ensemble de mesures 2D dans une séquence d'images. Les paramètres à estimer constituent l'état interne du système dynamique, résolu à travers un filtrage de Kalman étendu. Dans une version simplifiée [Cordea01], l'algorithme peut estimer uniquement la position 3D de la tête et la focale de la camera. D'autres développements [Strom99] utilisent l'approche *structure from motion* pour le codage à base de modèle.

Dans une démarche inspirée de la théorie de l'information, Viola et Wells [Viola95] proposent une technique générique d'estimation des paramètres de la pose 3D d'un objet rigide en maximisant l'information mutuelle calculée directement entre l'image et le modèle 3D projeté

[1] technique de mesure de longueurs et angles sur des photographies.

et en ignorant les phénomènes liés à la formation de l'image. L'optimisation stochastique est de nouveau utilisée. En conséquence, l'algorithme d'estimation est indépendant des conditions d'éclairement. Dans le cas du visage, le modèle est issu d'une scannérisation 3D du sujet. Les résultats numériques indiquant la précision de l'estimation ne sont pas fournis. En outre, les auteurs ne présentent pas une étude de la robustesse de l'estimation par rapport aux variations du modèle, celui-ci étant toujours relevé sur le sujet.

1.2.2 Méthodes locales

La démarche adoptée par ce type de méthodes consiste d'une part, à définir et extraire un ensemble de primitives représentatives associées aux éléments faciaux, et d'autre part, à interpréter la géométrie de ces primitives à l'aide d'un modèle 3D de visage.

Horprasert *et al.* [Horprasert96], dans l'hypothèse d'un suivi 2D sous-pixélique au niveau de cinq points remarquables du visage supposés invariants par rapport aux déformations (coins des yeux et extrémité du nez), développent un formalisme géométrique fondé sur des invariants projectifs et sur des données anthropométriques pour calculer trois angles définissant l'orientation 3D de la tête. Une étude sur la propagation des erreurs est en outre présentée.

Gee et Cipola [Gee94] considèrent en plus comme points rigides, les commissures des lèvres, ce choix étant discutable puisque la bouche est l'élément le plus déformable du visage. Un calcul géométrique élémentaire s'appuyant sur cette information augmentée permet d'exprimer l'orientation 3D de la tête et, de plus, la direction 3D du regard. L'étude de la propagation des erreurs est aussi présentée. En revanche, les détails de la procédure d'extraction des points d'intérêt ne sont pas fournis. Précisons que l'algorithme est capable d'effectuer le suivi 3D en temps réel même avec les moyens de calcul des années '90.

Plus orienté vers l'application et plus rigoureux en ce qui concerne l'extraction des primitives du visage, l'ouvrage de Stiefelhagen *et al.* [Stiefelhagen98] décrit un système complètement automatique capable de s'initialiser, de suivre en 3D le visage à une cadence vidéo de 15 images/s et de se remettre en marche suite à un échec. Dans une première étape de traitement, le visage est détecté grossièrement, en utilisant une modélisation gaussienne de la teinte de chair. Un seuillage adaptatif permet de localiser à l'intérieur du visage les pupilles et les bords des narines comme de petites taches ombrées, soumises à certaines contraintes de symétrie. Par la suite, la région de la bouche peut être prédite et l'analyse de ses contours conduit à la localisation des commissures des lèvres. L'algorithme rapide POSIT de DeMenthon et Davis [DeMenthon96] est appliqué pour les six points ainsi obtenus pour résoudre les correspondances 2D/3D. Testé sur des séquences de quelques centaines d'images, l'algorithme se montre stable, même si les erreurs

d'estimation des rotations dépassent parfois 20°.

En se plaçant à la limite de résolution de la correspondance 2D/3D, Colmenarez *et al.* [Colmenarez97] proposent une approche fondée sur la mise en correspondance entre trois points du visage et leur homologues sur un modèle 3D, notamment l'extrémité du nez et les coins extérieurs des yeux. Disposant d'un modèle 3D scanné, le suivi 2D des points d'intérêt s'effectue par un recalage 2D/3D/2D des éléments faciaux contenant ces points, en utilisant un filtre de Kalman. La même technique, appliquée sur d'autres éléments faciaux et couplée à un placage de texture, permet la création d'une séquence de synthèse incluant les déformations faciales. L'initialisation du modèle au début de la séquence s'effectue de manière automatique et consiste à rechercher des primitives de visage dans une base de prototypes préalablement constituée. Même si l'algorithme semble être bien mis au point, les auteurs n'indiquent pas les performances de l'estimation en terme d'erreur.

En se plaçant dans un cadre restreint d'estimation, Brunelli [Brunelli94] propose un algorithme permettant de localiser les yeux en utilisant des techniques simples d'analyse de gradients. Pour le rendre robuste, une première étape de traitement consiste à évaluer la direction de l'illumination afin de ramener l'image à une configuration standard. Les yeux ainsi localisés, permettent d'estimer la rotation latérale de la tête. Toutefois, il est nécessaire de fournir une initialisation, même grossière, de la position de l'œil.

1.3 Suivi des mouvements locaux du visage

Cette section traite des différentes méthodes proposées dans la litérature pour le suivi des mouvements locaux du visage dans des séquences vidéos monoscopiques. Dans une classification très générale, les techniques existantes peuvent être divisées, selon la méthodologie abordée, en deux grandes catégories. Les méthodes de la première catégorie, permettant l'estimation simultanée [Bozdagi94, Li93] ou successive [Black95, Choi94] des mouvements globaux et locaux du visage, ou même de la structure 3D [Azarbayejani95], ont été traitées dans la section 1.2. Elles sont principalement utilisées pour le codage vidéo à base de modèle 3D. Les techniques de la deuxième catégorie, présentées dans cette section, sont orientées plutôt vers l'analyse des mouvements des primitives du visage et elles ciblent un éventail plus diversifié d'applications. Ici, par primitive de visage nous entendons une région faciale d'intérêt pour une application donnée. En outre, par élément facial nous allons désigner explicitement une des régions correspondant : à la bouche, au nez, à l'œil et au sourcil.

Très généralement, un algorithme d'analyse des mouvements d'une primitive de visage s'articule autour de trois étapes de base :

- modéliser la primitive de visage et préciser les types des mouvements objets de l'analyse ;
- définir et extraire un ensemble de primitives d'image, capables de représenter de manière cohérente la primitive de visage considerée ;
- ajuster le modèle défini dans la première étape sur les primitives extraites de l'image.

L'étude de la littérature nous à conduit à une classification des techniques de capture des mouvements des primitives du visage structurée en deux niveaux. Le premier niveau renvoie aux différents choix concernant les primitives de visage. Le deuxième niveau prend en compte les possibilités de modélisation. La classification proposée est schématisée dans le Tableau 1.1 où sont indiquées les primitives d'image envisageables pour le suivi du mouvement. Nous allons détailler ensuite chacune de ces approches.

		Types de modèles					
		Sans modèle	Modèles rigides	Modèles déformables			
				Courbes	***Snakes***	**Prototypes déformables**	***Active shape models***
Types de primitives	Points	**FO**	**L**	**-**	**-**	**-**	**-**
	Régions	**L C PM**	**L**	**G L C**	**G FO**	**C PM**	**G L C**

Tableau 1.1 : Primitives d'image utilisables pour le suivi des mouvements locaux du visage (FO = flot optique, L = luminance, C = couleur, G = gradient, PM = primitive morphologique).

1.3.1 Suivi de points caractéristiques

Très peu adaptés à la représentation des détails du visage et partiellement justifiés par leur efficacité en temps de calcul, les modèles ponctuels permettent toutefois d'accomplir des tâches relativement sophistiquées d'analyse, telles que la reconnaissance du visage ou de l'expression faciale. Nous pouvons distinguer deux grandes approches de suivi de points caractéristiques : 1) suivi non-contraint et 2) suivi de points reliés.

1.3.1.1 Suivi non-contraint

La particularité de ces techniques de suivi consiste à donner toute liberté à l'ensemble de points caractéristiques définis de manière à obtenir un suivi indépendant de chacun de ces points.

Chon *et al.* [Lien00] proposent le suivi de 38 points du visage considérés comme pertinents pour la caractérisation de l'expression faciale en termes de *action units*. Typiquement, ces points sont placés sur les contours des lèvres, des yeux, des sourcils et du nez et extraits de manière interactive dans la première image de la séquence. Les points sont suivi indépendamment en utilisant l'estimateur local de mouvement de Lucas et Kanade [Lucas81], transposé dans un contexte multirésolution, pour prendre en compte les grands déplacements inter-trame. Le résultat est interprété en termes de *action units*, à l'aide d'un modèle de Markov caché pre-entraîné.

Tian *et al.* [Tian99], utilisant le même l'algorithme de Lucas et Kanade en y ajoutant un complément d'information relatif à la chromaticité du visage, proposent un algorithme capable d'effectuer, de plus, l'estimation de la configuration "ouvert/fermé" aussi bien pour la bouche que pour l'œil.

En dépit de la simplicité de mise en œuvre d'une modélisation non-contrainte, pour les techniques de suivi de points indépendants, deux remarques s'imposent :

- la disparition temporaire de certains points suite, par exemple, aux auto-occultations du visage, peut perturber fortement l'évolution du suivi,
- sans boucle de rétroaction (*feedback*), les erreurs risquent de se propager.

1.3.1.2 Suivi de points reliés

Le principe de ces techniques consiste à renforcer une certaine stabilité en introduisant des contraintes concernant les positions relatives des points d'intérêt.

Wang *et al.* [WangM98] sélectionnent 19 points représentatifs pour l'expression faciale, localisés au niveau des éléments faciaux, et expriment le suivi de ces points en terme d'optimisation d'un graphe étiqueté, dont les sommets sont les points d'intérêt et les arêtes interprétées sous forme de contraintes géométriques. Cela est exprimé par une fonction de coût regroupant un terme de corrélation inter-image au niveau des sommets et un terme de contrainte interne relative aux arêtes. Ce formalisme est d'ailleurs très similaire à celui utilisé par les prototypes déformables. L'optimisation de la fonction de coût est obtenue à l'aide d'un algorithme de recuit simulé et le résultat est utilisé à la reconnaissance de l'expression faciale. Des développements complémentaires de cette technique, appliqués à la reconnaissance du visage, se retrouvent dans [Pramadihanto98].

1.3.2 Suivi de régions d'intérêt

Une autre classe de primitives d'images utilisées dans le suivi des mouvements locaux du visage est celle des régions d'intérêt, exploitées dans un large spectre de techniques, qu'il s'agisse de suivi sans modèle géométrique, ou avec modèle rigide ou déformable. Ces techniques, telles qu'elles sont rencontrées dans la littérature, sont présentées par la suite.

1.3.2.1 Suivi sans modèle géométrique

Moyennant des hypothèses restrictives concernant le contenu et l'illumination de la scène, le suivi d'éléments faciaux sans disposer de modèles géométriques explicites est possible. Petajan et Graf [Petajan96] utilisent des techniques simples d'analyse colorimétrique, de seuillage et d'agrégation de pixels pour le suivi des bords des narines et des lèvres. Le résultat est utilisé pour animer un avatar capable de reproduire des mouvements simples de visage. Le traitement à base de seuillage est toutefois fortement limité par les conditions d'éclairement de la scène.

L'augmentation du contexte informationnel relatif à l'apparence de l'image dans les régions de certains éléments faciaux en terme de topographie (vallées locales, par exemple [WangR97]) peut apporter une relative indépendance par rapport à l'illumination.

En outre, des techniques simples de corrélation peuvent servir au suivi des éléments du visage. Machin [Machin96] propose la localisation des yeux en maximisant le coefficient de corrélation entre l'image et un prototype préétabli. Le clignement n'est en revanche pas pris en compte. La bouche est segmentée ensuite par seuillage adaptatif d'une région prédite à partir des positions des yeux. Le résultat est utilisé pour animer un modèle de synthèse en temps réel.

Antoszczyszyn *et al.* [Antoszczyszyn98], en s'inspirant du formalisme des *eigenfaces* [Turc91] introduit par l'équipe de Pentland et appliqué avec succès à la reconnaissance du visage, proposent une technique de suivi indépendant des yeux, de la bouche et du nez. Elle s'appuie sur une corrélation appliquée cette fois-ci dans les espaces résultant de l'analyse en composantes principales des régions d'image contenant ces éléments faciaux. L'analyse est effectuée sur un certain nombre d'images placées au début de la séquence et en même temps une base de prototypes est stockée. Le suivi de primitives de visage consiste à rechercher dans une zone d'intérêt le meilleur appariement, au sens des composantes principales, entre les prototypes et l'image. Cette approche conduit à une localisation pixélique des primitives considérées. Toutefois, l'analyse en composantes principales effectuée sur une partie très limitée de la séquence ne permet pas de capturer la statistique de toutes les configurations possibles.

1.3.2.2 Modèles rigides

Les modèles rigides ont la particularité d'imposer certaines contraintes entre les régions à géométrie fixe considérées comme primitives de visage dans le processus du suivi.

Par exemple, Zelinsky et Heinzmann [Zelinsky96] appliquent à l'analyse du visage l'approche de suivi de régions par le filtrage de Kalman. Ils utilisent un paquet de filtres pour suivre simultanément en 2D des régions restreintes placées sur les lèvres, les yeux et les sourcils. Les positions relatives de ces régions sont soumises aux contraintes géométriques de rigidité, ce qui permet de gérer les éventuelles auto-occultations de la tête. En outre, le système prend en compte la configuration "ouvert/fermé" de l'œil. Le résultat du suivi est utilisé dans une procédure de reconnaissance de geste, grâce à une décomposition des mouvements de la tête en actions élémentaires.

1.3.2.3 Modèles déformables

A l'encontre des modèles rigides, les modèles déformables utilisés comme primitives de visage peuvent changer de forme sous l'influence d'interactions internes (définies dans le modèle-même) ou externes (avec l'image). Ils sont ainsi capables d'assurer un suivi plus précis des mouvements locaux du visage. Comme montré au Tableau 1.1, nous avons structuré la classe de modèles déformables en plusieurs entités comme suit : 1) *lignes et courbes*, 2) *contours actifs*, 3) *prototypes déformables* et 4) *active shape models*.

1.3.2.3.a Lignes et courbes Schubert [Schubert00] utilise un polygone 3D rigide pour modéliser les contours extérieurs de l'œil. La transformation de Hough est un outil efficace permettant l'extraction des segments d'une image à partir de la carte des gradients, par une recherche de maxima locaux dans un domaine 2D limité. Toutefois, cette technique transposée à un modèle 3D linéaire et rigide projeté sur l'image, revient à la recherche d'un maximum global dans un espace 6-D (l'espace de paramètres du modèle - 3 rotations et 3 translations). En couplant cette approche à un prédicteur de Kalman, le temps de calcul est considérablement réduit, si bien que l'algorithme fonctionne en temps réel. L'inconvénient principal de cette méthode renvoie à l'instabilité des gradients dans la région de l'œil. De plus, le modèle rigide ne peut pas prendre en compte les clignements.

La modélisation par splines offre l'avantage d'une représentation paramétrique compacte pour des objets relativement complexes tout en gardant la précision d'approximation. Cela la rend particulièrement intéressante en analyse d'images quand il s'agit d'accomplir des tâches d'optimisation dans l'espace des paramètres du modèle.

Moses *et al.* [Moses95] utilisent une B-spline ouverte pour le suivi de la région inter-lèvres. La particularité de leur approche réside dans l'utilisation d'un filtre de Kalman conçu pour modéliser la dynamique du contour inter-lèvre. Cela confère robustesse et permet le traitement en temps réel. Evidement, l'approche prend en compte uniquement la configuration "fermé" de la bouche. Un classificateur bayesien utilise le résultat du suivi pour discriminer entre cinq configurations associées aux différentes expressions faciales.

Sanchez *et al.* [Sanchez97] utilisent un modèle réaliste de lèvres à base de B-splines contrôlés par 10 points. Le suivi est effectué dans un schéma de décision statistique fondée sur une modélisation de la chromaticité dans la région de la bouche sous forme de mélange gaussien à deux composantes (tente de chair/lèvre). Toutefois, cette modélisation pourrait être trop limitative pour gérer, par exemple, la chromaticité des dents.

1.3.2.3.b Contours actifs Les contours actifs, ou *snakes* [Kass87], présentent une formulation énergétique pour des objectifs de segmentation non-rigide en vision par ordinateur. En général, la segmentation d'une région de l'image de forme complexe par une modélisation à base d'objets géométriques paramétrés (courbes dans le cas 2D ou surfaces en 3D) revient à un problème mal posé. Les *snakes* introduisent des contraintes de régularité dans l'espace de la solution recherchée, par l'intermédiaire de potentiels intrinsèquement associés au modéle et dérivés des formulations mécaniques. Pour exposer très brièvement le formalisme des *snakes*, notons $\mathcal{M}(\mathbf{p})$ le modèle géométrique paramétré sur un espace $\mathcal{P} \ni \mathbf{p}$ et I l'image contenant l'objet d'intérêt à segmenter. L'approche de segmentation par *snakes* consiste à définir deux fonctionnelles énergétiques $E_{\text{int}}(\mathcal{M}(\mathbf{p}))$ et $E_{\text{ext}}(\mathcal{M}(\mathbf{p}), I)$ et de résoudre :

$$\widehat{\mathbf{p}} = \underset{\mathbf{p}\in\mathcal{P}}{\arg\min}\ [E_{\text{int}}(\mathcal{M}(\mathbf{p})) + E_{\text{ext}}(\mathcal{M}(\mathbf{p}), I)]\ .$$

Le premièr terme, E_{int}, nommé énergie interne, formalise les propriétés *a priori* du modèle en termes de contraintes élastiques et joue le rôle de terme régularisant pour la solution. L'énergie externe, E_{ext}, exprime l'interaction du modèle avec les données de l'image. Les différentes approches de segmentation à base de *snakes* se distinguent par :

- la représentation analytique du modèle : courbe/surface explicite (paramétrée) ou implicite sur un espace fonctionnel non-paramétrique (par exemple fonctions presque partout $\mathcal{C}^k$ sur $\mathbf{R}^n$) ou paramétrique (développement en série avec fonctions de base - Fourier, splines, etc.) ;
- la nature de la fonctionnelle de régularisation E_{int} : membrane, plaque-mince, splines à continuité contrôlée, etc. ;

- la nature de la fonctionnelle d'attache aux données, E_{ext} : celle-ci dépend du problème à résoudre. De manière générale, elle renvoie aux équations de contrainte du problème considéré (par exemple maximalité de $\|\nabla I\|$ dans le cas de la segmentation orientée contours, équation de contrainte du flot optique pour une segmentation orientée mouvement) et aux primitives d'image associées ;
- la méthode d'estimation de $\widehat{\mathbf{p}}$: minimisation directe de l'énergie par un algorithme d'optimisation (programmation dynamique, algorithmes génétiques) ou approches variationelles (descente de gradient, dynamique lagrangienne, etc.).

Les *snakes* sont considérés comme des outils génériques et puissants de segmentation automatique en vision par ordinateur. Rappelons toutefois leurs deux points faibles : la robustesse du résultat dépend fortement de l'initialisation et leur coût élevé en temps de calcul.

En ce qui concerne l'imagerie faciale, les snakes peuvent être aussi bien utilisés pour de simples tâches de segmentation, que pour la capture des mouvements 2D/3D non-rigides du visage. Fukuhara et Murakami [Fukuhara93] utilisent les *snakes* pour segmenter le visage entier, la bouche, les yeux et le nez dans leur système de codage à base de maillage 3D. Pardas [Pardas00] propose un algorithme capable de détecter automatiquement les yeux et de les suivre, par estimation de mouvement et segmentation à base de *snakes*, par rapport aux contours extérieurs. Son algorithme peut même gérer les clignements.

Terzopoulos et Waters [Terzopoulos93] utilisent les contours actifs pour extraire certaines primitives du visage à des fins d'analyse et de synthèse dynamique des expressions faciales. Le résultat de la segmentation des primitives faciales d'intérêt (lèvres, sourcils, bords des cheveux et du menton), dont la robustesse est renforcée par l'utilisation de marqueurs, est injecté dans un modèle dynamique simulant les contractions musculaires et les déformations de la peau, dérivé du concept de *Facial Action Coding System* de Eckman et Friesen [Ekman77]. L'avantage de cette approche est le rendu réaliste de l'image de synthèse.

Wang et Lee [WangY94] introduisent le concept de maillage actif et démontrent son utilité pour des applications de suivi d'objet et interpolation. La méthode est fondée sur la modélisation de l'image par un maillage 2D (quadrangulaire) déformable. Les pixels situés à l'intérieur des facettes sont interpolés à partir des valeurs de luminance des sommets du maillage. Les déplacements des sommets sont calculés entre deux trames successives d'une séquence et spécifient la déformation du maillage. Afin de déterminer les champs de déplacements, les auteurs utilisent une simple descente de gradient pour minimiser une fonctionnelle d'énergie regroupant les quatre termes suivants :

- une énergie interne de déformation du maillage, introduite comme contrainte de régularité,

- une énergie "d'interpolation", mesurant le degré d'uniformité de l'image à l'intérieur des facettes du maillage,
- une énergie de "primitives", nécessaire afin de forcer le positionnement des sommets du maillage sur les gradients de forte amplitude de l'image, et
- une énergie "d'ajustement", incorporant l'erreur de compensation de mouvement associée aux champs de déplacement entre deux trames successives.

Cette construction est bien adaptée pour des applications d'interpolation et de compensation du mouvement dans le contexte de la compression des séquences vidéos, mais la granularité du maillage (32x32 pour des images de taille 256x256 et plus) empêche un suivi très précis des primitives du visage.

1.3.2.3.c Prototypes déformables Les prototypes déformables (*deformable templates*) [Yuille92] sont très similaires aux contours actifs au niveau de la formulation en termes d'énergies interne et externe, mais ils incorporent plus d'information *a priori* sur la structure de l'objet ciblé, notamment pour exprimer l'interaction modèle-image. Les prototypes déformables sont des outils dédiés à des applications spécifiques et, en général, plus robustes que les *snakes* car ils exploitent de manière précise l'apparence de l'image dans la région ciblée.

La plupart des approches fondées sur les prototypes déformables reprennent la formulation introduite par Yuille *et al.* [Yuille92]. L'apport des approches existantes se limite à la modification des termes énergétiques ou à l'ajout de nouveaux termes afin de rendre plus robuste le recalage.

S'imposant progressivement, la modélisation par prototypes déformables a été utilisée avec succès en imagerie faciale appliquée au codage à base de modèle 3D [Reinders95, Zhang.L98], ce qui a suscité notre intérêt pour ce type d'approche. Les prototypes déformables seront présentés en détails dans la section 4.2.1.

1.3.2.3.d *Active shape models* Les *active shape models* [Cootes95] proposent une approche probabiliste pour l'apprentissage et l'estimation des déformations d'objets discrétisés, en terme d'analyse en composantes principales. Plus précisément, soit $\mathbf{x}_i = (x_i, y_i),\ i = 1, 2, \ldots, n$ un ensemble de points de contours de l'objet à analyser. Dans ce cas, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ donne une représentation paramétrique de l'objet dans $\mathbf{R}^{2n}$. Considérons $\mathbf{X}$ comme une réalisation d'un vecteur aléatoire $\mathbf{\Psi}$. Mesurer $\mathbf{X}$ sous différentes hypotèses de déformation de l'objet permet l'analyse en composantes principales de $\mathbf{\Psi}$ pour réduire la dimension de l'espace de paramétrisation. Cela est effectué sur une base d'apprentissage expérimentalement établie. L'estimation de la forme de l'objet contenu dans une image consiste en une étape d'optimisation,

effectuée dans l'espace réduit de paramétrisation, pour l'appariement entre $\mathbf{X}$ et les données de l'image. Pour cela, des algorithmes utilisant la carte de gradients [Cootes94] ou mettant en œuvre des modélisations statistiques pendant l'étape d'apprentisage [Haslam94, Luettin96] ont été proposés. Notons que la création de la base d'apprentissage est une étape laborieuse, comportant le choix des modes de déformation ainsi que l'alignement des objets par rapport à la rotation et à l'échelle. Cela constitue le principal inconvénient de cette approche.

Luettin *et al.* [Luettin96] utilisent les *active shape models* pour le suivi du mouvement des lèvres. Ils proposent une étape d'apprentissage comportant l'analyse en composantes principales sur la carte des gradients au niveau des pixels situés dans un voisinage des contours des lèvres. Cela est effectué sur une base d'images représentant des interlocuteurs provenant de différentes ethnies (présentant donc des caractéristiques faciales différentes) mais reproduisant la même série de mots. L'estimation est effectuée en minimisant par descente du simplexe une fonction de coût quadratique, définie à partir des composantes principales des gradients. Selon un critère subjectif d'évaluation, le suivi précis est obtenu dans 98% des cas. Lucey *et al.* [Lucey00] améliorent cette technique en introduisant un module supplémentaire d'initialisation, mettant en œuvre le concept de *eigenlip* [Bregler94]. De plus, une représentation multiéchelle à base d'ondelettes peut améliorer la robustesse par rapport aux variations des conditions d'éclairement [Xu98].

On peut également considérer l'information colorimétrique associée aux lèvres. Sum *et al.* [Sum01] proposent un modèle prenant en compte uniquement les contours extérieurs des lèvres, dont l'ajustement s'effectue sur une carte de segmentation issue d'une classification floue de la couleur des pixels situés dans une région d'intérêt. Cela conduit à un pourcentage de suivi correct légèrement inférieur, de 95%.

Disposant d'un maillage 3D et d'un placage de texture exprimé comme une combinaison linéaire de *eigenfaces*, l'*active appearance model* [Cootes98] généralise le concept de *active shape model*. L'utilisation de l'*active appearance model* pour l'adaptation du maillage directement sur l'image dans un contexte de codage de visage MPEG-4 à base de modèle 3D a été récemment rapporté dans [Ahlberg01]. La phase d'apprentissage du modèle reste toutefois la limitation principale de cette approche.

Des approches combinées, intégrant plusieurs représentations en termes de modèles déformables ont été aussi envisagées. Chiou et Hwang [Chiou97] développent un système de suivi des lèvres mettant en œuvre les *snakes* et les *active shape models* dans un contexte de traitement des images couleurs. Le résultat est appliqué pour la lecture sur les lèvres par reconnaissance temporelle à base de modèles de Markov cachés.

1.4 Les approches adoptées et développées

Dans le contexte des applications de capture de mouvements, nous proposons une approche robuste à base de modèle pour le suivi du visage dans des séquences vidéos acquises dans un contexte réaliste, visant à contrôler les conditions suivantes : 1) acquisition avec une seule caméra non calibrée, fixe ou mobile, 2) conditions non stabilisées d'éclairement, 3) mouvements complexes de grande amplitude, 4) déformations locales liées aux expressions faciales, 5) occultations partielles de la tête.

S'inscrivant dans la ligne classique de l'analyse de séquences vidéos faciales, l'approche proposée comporte deux étapes de difficulté croissante : 1) une adaptation globale de la pose du modèle, en assimilant le mouvement de la tête à un mouvement rigide, 2) une adaptation locale de la forme du modèle, pour prendre en compte les déformations/mouvements des différentes parties du visage.

Le chapitre 2 décrit une méthode générique d'estimation de la pose 3D globale de la tête dans des séquences vidéos monoscopiques, non-calibrées, à travers une mise en correspondance de primitives 3D d'un modèle de tête avec des primitives 2D extraites des images.

Tout d'abord, une méthode de synthèse d'un modèle générique de tête par une approche analytique est présentée et son adéquation à notre problématique est discutée par rapport aux techniques de représentation par maillage.

Les ensembles de primitives 3D du modèle et 2D des images sont ensuite définis. Ces primitives renvoient à des informations de mouvement, au travers du champ de déplacements, et de texture, via la photométrie. Le choix d'une méthode d'estimation du champ de déplacements nous a conduit à analyser en termes de performance et de précision les approches différentielles intégrées dans un schéma multirésolution et celles de type appariement par bloc en contexte monorésolution. De cette analyse comparée, nous avons opté pour une approche d'estimation du flot optique par programmation dynamique orthogonale. Nous avons déduit ensuite le concept de flot coloré par introduction des informations de texture dans la région de la tête et nous avons mis en œuvre les primitives définies pour réaliser le recalage 3D/2D du modèle de tête sur les images. Le principe de la mise en correspondance repose sur la minimisation par la méthode du simplexe d'une fonctionnelle d'erreur composite, définie de manière à assurer la propagation d'une image à l'autre de l'information liée à la pose 3D du modèle, à travers le flot coloré.

Afin de pallier les limitations bien connues d'estimation du flot optique, une compensation en mouvement translationnel dominant a été introduite en utilisant une approche de type appariement par blocs.

En dépit d'une bonne précision générale de recalage, les résultats obtenus sur des séquences synthétiques et réelles montrent toutefois des décrochements survenus en présence de mouvements complexes de grande vitesse angulaire générant des auto-occultations et des occultations occasionnelles.

Le chapitre 3 propose des apports méthodologiques pour s'affranchir des limitations de la méthode d'estimation de la pose 3D de la tête, en introduisant le concept de flot coloré.

Ainsi, utilisant le champ de déplacements compensé en translation dominante, nous avons développé une technique d'interpolation spatio-temporelle fondée sur une modélisation ondulatoire hiérarchique dans le cadre d'une approche physique par groupe de paquets d'onde. Cette interpolation nous a permis de synthétiser des images virtuelles afin de guider un recalage 3D/2D stable et précis en présence de grandes rotations. De plus, pour mieux gérer les auto-occultations de la tête, nous avons introduit un critère de visibilité issu des principes d'estimation robuste.

Nous avons montré comment les cas d'occultation au sens du mouvement peuvent être gérés en considérant une classification fondée sur un critère de mouvement sous contrainte de régularité spatiale, couplée à une analyse de similarité de mouvements à base de modèle paramétrique.

Concernant la capture des mouvements non rigides du visage et des expressions faciales, le chapitre 4 présente une méthode de recalage à base de prototypes déformables pour le suivi de la bouche et des yeux dans des séquences vidéos monoscopiques contenant un visage parlant.

Les prototypes proposés, conçus de manière à intégrer les descripteurs de visage définis par le standard MPEG-4, sont obtenus dans le cadre d'une représentation souple et précise, à l'aide de B-splines. Ils sont intrinsèquement caractérisés par des contraintes élastiques et de symétrie locale héritées d'une modélisation physique à base de ressorts. Dans le cas particulier de l'œil, deux prototypes ont été proposés, correspondant respectivement à la configuration ouverte/fermée de celui-ci.

Les déformations auxquelles les prototypes sont soumis par interaction avec les données images sont ensuite spécifiées. Typiquement, des primitives de gradient et de texture, combinées à une carte de segmentation floue sous contrainte spatiale sont mises en œuvre, de manière à exploiter les caractéristiques de chaque élément facial considéré.

Les contraintes internes et externes des prototypes sont combinées dans un schéma d'optimisation par la méthode du simplexe. L'initialisation robuste des prototypes d'une image à l'autre est effectuée en couplant une procédure de segmentation automatique de l'iris et de détection de la configuration ouverte/fermée de l'œil à l'estimation de la pose 3D de la tête.

Cet ensemble s'achève sur une conclusion générale, une discussion de voies méthodologiques à explorer pour améliorer encore l'approche développée et sur l'esquisse d'orientations de recherche et leurs perspectives dans un contexte générique de suivi d'objet 3D articulé et déformable.

Chapitre 2

Estimation de la pose 3D du visage par modèle d'objet

2.1 Positionnement du problème

La principale difficulté à laquelle sont confrontées les techniques de codage vidéo orientées modèle ou la création de scènes 3D animées virtuelles ou augmentées, réside dans l'estimation et le suivi de la pose 3D d'objets dans des environnements complexes. La précision de l'estimée de pose conditionne en effet la robustesse de la plupart des applications référencées vision, qu'il s'agisse du contrôle en temps réel du trafic urbain, routier ou autoroutier [Koller93, Kollnig97], de l'analyse de geste [Rohr97] (sportif ou pour les interfaces homme/machine), de la commande visuelle de robot [Horaud98], de l'animation de systèmes 3D [Abrantes97], ou de l'estimation de la forme d'un objet [Metaxas93].

La pose 3D d'un objet est définie par les paramètres globaux du modèle (rotations, translations et facteur d'échelle) garantissant la mise en correspondance de primitives d'un modèle 3D de l'objet projetées sur l'image, avec les primitives 2D analogues dans l'image. En particulier, l'estimation de la pose 3D d'un visage est un problème difficile d'analyse, en raison :

- de la géométrie spécifique de la tête (objet non convexe présentant de fortes variations de courbure),
- des mouvements complexes susceptibles d'être produits,
- des déformations locales plus ou moins prononcées, liées à la richesse des expressions faciales,
- de la diversité des contenus correspondant à des scènes complexes pour des conditions d'éclairement *a priori* inconnues et non stabilisées.

L'approche la plus classique mettant en œuvre un modèle d'objet de tête pour analyser des séquences vidéos faciales [Aizawa95, Bozdagi94, Li93, Zhang.L98] comporte deux étapes de difficulté croissante : 1) une adaptation globale de la pose du modèle, en assimilant le mouvement de la tête à un mouvement rigide, 2) une adaptation locale de la forme du modèle, pour prendre en compte les déformations/mouvements des parties du visage les plus expressives dans une communication (bouche, yeux, sourcils).

En adoptant une telle approche, nous avons développé une méthode robuste d'estimation de la pose 3D globale de la tête dans des séquences vidéos acquises dans un contexte réaliste, visant à contrôler les conditions suivantes :

- acquisition avec une seule caméra non calibrée, fixe ou mobile,
- conditions non stabilisées d'éclairement,
- scènes d'intérieur ou d'extérieur,
- mouvements complexes de grande amplitude,
- déformations locales liées aux expressions faciales,
- occultations partielles de la tête.

Tout d'abord, nous présentons et discutons les approches méthodologiques et les outils algorithmiques dont nous avons besoin pour mettre en œuvre une estimation robuste de la pose 3D, fondée sur l'utilisation du flot optique. Dans la suite, après avoir présenté la méthode de synthèse d'un modèle générique de tête par une approche analytique et discuté son adéquation à notre problématique par rapport aux techniques de représentation par maillage, nous définissons les primitives 3D du modèle et 2D des images prises en compte. Ces primitives correspondent à des informations de mouvement au travers du champ de déplacement et de texture via la photométrie.

Nous rappelons les hypothèses sur lesquelles reposent l'estimation du flot optique et analysons en termes de performance et de précision les approches différentielles intégrées dans un schéma multirésolution et celles de type appariement par bloc en contexte monorésolution. De cette analyse comparée et des résultats expérimentaux obtenus, nous justifions le choix d'une approche d'estimation du flot par programmation dynamique orthogonale. Nous en déduisons alors le concept de flot coloré par inégration des informations de texture dans la région de la tête.

Après avoir introduit le principe de compensation du mouvement translationnel dominant, afin de pallier les limitations bien connues d'une estimation du flot optique, nous mettons en œuvre les primitives définies pour réaliser le recalage 3D/2D du modèle de tête sur les images.

Le principe de la mise en correspondance est détaillé, les deux composantes d'erreur introduites sont justifiées et leur minimisation est effectuée par la méthode du simplexe. Enfin, les résultats obtenus sur des séquences synthétiques et réelles sont discutés et les cas d'échec analysés.

2.2 Techniques de base

2.2.1 Estimation du mouvement apparent de la scène : flot optique

2.2.1.1 Généralités

Un sujet fondamental en traitement des séquences vidéos consiste à estimer le champ de vitesses 2D résultant de la projection du mouvement 3D de la scène. Formellement, ce champ est défini par :

$$\mathbf{v} = \left(\frac{dx}{dt}, \frac{dy}{dt}\right), \ (x, y) = (\pi \circ c_t \circ m_t)(X, Y, Z), \ \ (X, Y, Z) \in \mathcal{S} \ ,$$

où :

- $\mathcal{S}$ représente l'ensemble des points 3D de la scène dans un système de coordonnées fixé,
- $m_t : \mathbf{R}^3 \to \mathbf{R}^3$ décrit le mouvement présent dans la scène au temps t,
- $c_t : \mathbf{R}^3 \to \mathbf{R}^3$ permet de passer au système de coordonnées (mobile) de la caméra à l'instant t,
- $\pi : \mathbf{R}^3 \to \mathbf{R}^2$ désigne la transformation projective effectuée par le système optique de l'objectif de la caméra.

En pratique, en raison de l'échantillonnage temporel, le champ de vitesses est approché par un champ de déplacements entre les trames adjacentes, en remplaçant les dérivées par des différences finies. Le champ de déplacements dépend :

- de la structure 3D de la scène (composition et orientation des objets),
- du mouvement présent dans la scène,
- du mouvement de la caméra,
- du système optique de l'objectif.

Le calcul du champ de déplacements est généralement connu sous le terme d'estimation du mouvement apparent de la scène. Remarquons que nous ne pouvons accéder au mouvement apparent de la scène que par l'intermédiaire des variations temporelles perçues dans les images fournies par la caméra. En général, le mouvement 2D estimé dans l'image, connu sous le nom

de flot optique, diffère du mouvement apparent de la scène. Toutefois, en pratique, ces deux notions sont considérées comme identiques.

En conclusion, trois facteurs contribuent à l'ambiguïté du mouvement 3D de la scène par rapport au mouvement 2D observé dans l'image :

- les mouvements de la caméra,
- la perte d'information due à la projection 3D/2D,
- l'altération du mouvement 2D perçu due au mécanisme de la formation de l'image sur le capteur de la caméra.

Néanmoins, le flot optique se présente comme un descripteur riche et générique de mouvement, envisageable dans une large classe d'applications en vision par ordinateur. La connaissance du mouvement 2D permet de remonter à des informations sur la dynamique 3D de la scène. En outre, certaines approches, initiées par les travaux de Hay [Hay66] et Longuet-Higgins et Prazdny [Longuet-Higgins80] et connues sous la terminologie de *structure from motion* tendent à inférer la structure 3D de la scène à partir du champ de déplacements 2D.

2.2.1.2 Méthodes de calcul

Traditionnellement, l'estimation du flot optique a été formulée sous la contrainte de conservation/stationnarité locale de la luminance. Des approches plus récentes relâchent cette contrainte dans des formulations portant sur les effets photométriques liés à la formation de l'image. Dépassant le cadre de nos recherches, ces approches ne sont pas détaillées dans ce mémoire. Toutefois, le lecteur intéressé peut se référer à [Verri89] ou [Pentland91].

Concernant la formulation de l'estimation du flot optique, nous pouvons distinguer deux types d'approches :

- les techniques non-paramétriques : les vecteurs de déplacement sont estimés localement, en chaque pixel de l'image ;
- les techniques paramétriques : les déplacements des pixels sont agrégés dans des modèles paramétriques de mouvement et l'estimation est effectuée dans l'espace de paramètres associé.

En outre, les méthodes d'estimation du flot optique peuvent être divisées en trois catégories, en fonction des hypothèses de départ et de la méthode de calcul retenue :

- les méthodes différentielles,
- les méthodes par corrélation,

- les méthodes fréquentielles.

Les approches fréquentielles sont plus restrictives au niveau des applications, car elles supposent une nature localement translationnelle du mouvement 2D présent dans l'image. Par conséquent, les deux premières catégories de méthodes sont les plus couramment utilisées en pratique. Dans cette section, nous détaillons et illustrons ces deux types d'approche. En outre, nous présentons une analyse qualitative et quantitative effectuée sur trois algorithmes d'estimation du flot optique, à partir des algorithmes de Horn et Schunk, de Lucas et Kanade et de Quénot.

2.2.1.2.a Méthodes différentielles : l'algorithme de Horn et Schunk Les méthodes différentielles de calcul du flot optique ont pour origine l'algorithme de **Horn et Schunk** [Horn81], dont la contrainte de conservation de la luminance I de l'image est exprimée sous la forme d'une dérivée temporelle nulle :

$$\frac{d}{dt} I(x, y, t) = 0 \ , \tag{2.1}$$

où x et y sont les coordonnées spatiales dans le plan de l'image. Considérant des conditions suffisantes de régularité, l'équation (2.1) s'écrit sous la forme :

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \ ,$$

où sous la forme compacte :

$$\nabla I \cdot \mathbf{v} = -I_t \ , \tag{2.2}$$

avec :

$\nabla I = (I_x,\, I_y) \overset{\text{not}}{=} \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right)$, $\mathbf{v} = \left(\frac{dx}{dt}, \frac{dy}{dt}\right) \overset{\text{not}}{=} (u, v)$, le vecteur vitesse, à calculer, et $I_t \overset{\text{not}}{=} \frac{\partial I}{\partial t}$.

L'equation (2.2) porte le nom de d'Equation de Contrainte du Mouvement Apparent (ECMA). Remarquons que la condition (2.1) est plus restrictive que la contrainte de conservation locale à court terme de la luminance, notamment :

$$I(x + \delta x,\, y + \delta y,\, t + \delta t) = I(x, y, t) \ . \tag{2.3}$$

Cette dernière conduit, par un développement limité de Taylor au premier ordre, à :

$$\frac{\partial I}{\partial x}\frac{\delta x}{\delta t} + \frac{\partial I}{\partial y}\frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \approx 0 \ ,$$

ce qui donne, à la limite ($\delta t \to 0$), une forme approximative de la contrainte (2.2). De plus, notons que (2.2) introduit une incertitude au niveau du vecteur vitesse sur les zones d'intensité constante.

Seule, l'équation (2.2) ne permet pas de déterminer complètement le vecteur vitesse. Elle fournit une seule composante de ce vecteur, via la projection sur la direction du gradient spatial de l'image, $-I_t/\sqrt{I_x^2 + I_y^2}$. Pour pouvoir calculer le vecteur vitesse complet, il est nécessaire de formuler des hypothèses supplémentaires. Horn et Schunk introduisent une contrainte de lissage spatial, suite à l'observation qu'en général le champ de vitesse présente des propriétés de régularité locale. Dans certaines situations, par exemple, au voisinage des contours occultants, cela n'est plus valable. Explicitement, cette contrainte de lissage est injectée dans une fonction de coût composite, de la forme :

$$E = \iint_{\mathcal{D}} \left(\varepsilon_{\mathrm{c}}^2 + \alpha^2 \varepsilon_{\mathrm{r}}^2\right) \, dxdy \ ,$$

où

- $\varepsilon_{\mathrm{c}} = \nabla I \cdot \mathbf{v} + I_t = I_x \, u + I_y \, v + I_t$ est le terme provenant de la contrainte différentielle (2.2),
- $\varepsilon_{\mathrm{r}}^2 = \nabla^2 u + \nabla^2 v = \left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) + \left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right)$ est le terme de régularisation, limitant la variation locale du vecteur vitesse par l'intermédiaire de son Laplacien.
- $\mathcal{D}$ représente le support de l'image où le champ de vitesses est estimé,
- α est un paramètre contrôlant le lissage.

Le calcul du champ de vitesses revient à minimiser E par rapport aux fonctions u et v. Dans le cas discret (échantillonnage spatial), le Laplacien est approché par :

$$\begin{aligned} \nabla^2 f\,(x,\,y) &\approx \overline{f}\,(x,\,y) - f\,(x,\,y)\,, \text{ avec} \qquad (2.4) \\ \overline{f}\,(x,\,y) &= \frac{1}{6}\,[f\,(x-1,\,y) + f\,(x+1,\,y) + f\,(x,\,y-1) + f\,(x,\,y+1)] + \\ &\quad \frac{1}{12}\,[f\,(x-1,\,y-1) + f\,(x-1,\,y+1) + f\,(x+1,\,y-1) + f\,(x+1,\,y+1)] \end{aligned}$$

et le problème variationnel issu de la minimisation de E est réduit à la résolution d'un système linéaire :

$$\left\{ \begin{array}{l} 0 = \dfrac{\partial E}{\partial u}\,(x,y) = \left[2\,(I_x\,u + I_y\,v + I_t)\,I_x + 2\,\alpha^2\,(u - \overline{u})\right]\big|_{(x,\,y)} \\ 0 = \dfrac{\partial E}{\partial v}\,(x,y) = \left[2\,(I_x\,u + I_y\,v + I_t)\,I_y + 2\,\alpha^2\,(v - \overline{v})\right]\big|_{(x,\,y)} \end{array} \right. , \ (x,\,y) \in \mathcal{G}\ , \qquad (2.5)$$

$\mathcal{G}$ étant l'ensemble de points de la grille d'échantillonnage. La limitation de cette approche réside dans la taille importante du système (2.5), même pour des images de dimensions modestes. Pour

s'affranchir de cet inconvénient, Horn et Schunk proposent une solution itérative de type Gauss-Seidel :

$$\begin{array}{l} \widehat{u}_0 = \widehat{v}_0 = 0 \\ \widehat{u}_{k+1} = \overline{\widehat{u}}_k - I_x \left(I_x \, \overline{\widehat{u}}_k + I_y \, \overline{\widehat{v}}_k + I_t\right)/(\alpha^2 + E_x^2 + E_y^2) \\ \widehat{v}_{k+1} = \overline{\widehat{v}}_k - I_y \left(I_x \, \overline{\widehat{u}}_k + I_y \, \overline{\widehat{v}}_k + I_t\right)/(\alpha^2 + E_x^2 + E_y^2) \end{array} \quad , \; k \geq 0 \; . \tag{2.6}$$

Les équations (2.6) peuvent être interprétées comme une succession d'étapes de filtrage passe-bas, effectuées jusqu'à ce qu'un critère de convergence soit satisfait. Les détails d'implantation de cet algorithme seront présentés dans la section 2.2.1.4.

Comme développement ultérieur de l'algorithme de Horn et Schunk, rappelons l'approche de Nagel et Enkelmann [Nagel86], qui adoptent la solution d'un lissage anisotrope, diminué dans la direction des variations fortes de la luminance (direction perpendiculaire aux contours de l'image), avec un formalisme utilisant des dérivées du deuxième ordre. Cette technique est capable de mieux gérer les discontinuités du mouvement générées par les occultations, mais implique une augmentation non négligeable du volume de calcul. Néanmoins, des expérimentations effectuées sur des séquences standard de test, n'indiquent qu'une amélioration médiocre au niveau de la précision de l'estimation, par rapport à l'algorithme de Horn et Schunk [Barron94].

2.2.1.2.b Méthodes par corrélation : les algorithmes de Lucas et Kanade et de Quénot Ce type de méthodes s'appuie sur la recherche des correspondances entre des paires de régions extraites de deux trames d'une séquence. L'approche la plus triviale pour estimer le déplacement d'un pixel (x, y) entre deux images, I et J, consiste à sélectionner un bloc $B_{(x,\,y)}$ dans l'image I, contenant le pixel (x, y), et à trouver son correspondant dans l'image J, par une recherche exhaustive dans un voisinage de (x, y) :

$$(u, v) = \underset{\left(\substack{0 \leq \triangle x \leq \triangle x_{\max} \\ 0 \leq \triangle y \leq \triangle y_{\max}}\right)}{\arg\min} \sum_{(x,\,y) \in B_{(x,\,y)}} \left| I(x, y) - J(x + \triangle x, y + \triangle y) \right| \; .$$

Cette technique d'appariement par bloc est utilisée, par exemple, dans les modules de compensation du mouvement des codeurs MPEG-1 et MPEG-2. Son principal inconvénient réside dans le risque de faux appariements, d'autant plus important qu'augmente la taille du voisinage de recherche.

Lucas et Kanade [Lucas81] proposent une procédure itérative pour résoudre l'appariement par bloc au sens des moindres carrés, leur algorithme se situant parmi les plus performant en terme de précision d'estimation du flot optique [Barron94]. Avec les notations précédentes, le calcul du vecteur de déplacement au pixel $\mathbf{x} = (x, y)$ revient à minimiser :

$$E(\mathbf{v}) = \sum_{\mathbf{y} \in B_{\mathbf{x}}} \left[I(\mathbf{y}) - J(\mathbf{y} + \mathbf{v}) \right]^2 \; .$$

Sous l'hypothèse de petits déplacements, on utilise l'approximation :

$$J(\mathbf{y}+\mathbf{v}) \approx \nabla J(\mathbf{y}) \cdot \mathbf{v} \tag{2.7}$$

et en imposant $0 = \dfrac{\partial E}{\partial u} = \dfrac{\partial E}{\partial v}$, le vecteur de déplacement s'exprime comme la solution du système linéaire :

$$\sum_{\mathbf{y} \in B_{\mathbf{x}}} \left[I(\mathbf{y}) - \nabla J(\mathbf{y}) \cdot \mathbf{v} \right] \nabla J(\mathbf{y}) = \mathbf{0} \ ,$$

avec :

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left[\dfrac{\partial J}{\partial x}(\mathbf{y}) \right]^2 & \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \dfrac{\partial J}{\partial x}(\mathbf{y}) \ \dfrac{\partial J}{\partial y}(\mathbf{y}) \\ \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \dfrac{\partial J}{\partial x}(\mathbf{y}) \ \dfrac{\partial J}{\partial y}(\mathbf{y}) & \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left[\dfrac{\partial J}{\partial y}(\mathbf{y}) \right]^2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} I(\mathbf{y}) \ \dfrac{\partial J}{\partial x}(\mathbf{y}) \\ \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} I(\mathbf{y}) \ \dfrac{\partial J}{\partial y}(\mathbf{y}) \end{pmatrix} . \tag{2.8}$$

Notons que l'expression (2.8) donne une solution approchée du vecteur de déplacement car elle utilise (2.7). La précision d'estimation pourrait être améliorée si on dispose d'un pré-estimé $\widehat{\mathbf{v}} = (\widehat{u}, \widehat{v})$ du vecteur de déplacement. Dans ce cas, (2.7) devient :

$$J(\mathbf{y} + \widehat{\mathbf{v}} + \mathbf{v}) \approx \nabla J(\mathbf{y} + \widehat{\mathbf{v}}) \cdot \mathbf{v}$$

et (2.8) garde la même forme, sauf que les dérivées $\dfrac{\partial J}{\partial x}$ et $\dfrac{\partial J}{\partial y}$ seront calculées au point $\mathbf{y} + \widehat{\mathbf{v}}$. Cette observation suggère la formule itérative suivante pour une estimation précise du vecteur de déplacement au pixel $\mathbf{x}$:

$$\begin{aligned} \widehat{\mathbf{v}}_0(\mathbf{x}) &= 0 \ , \\ \widehat{\mathbf{v}}_{k+1}(\mathbf{x}) &= \widehat{\mathbf{v}}_k(\mathbf{x}) + \begin{pmatrix} \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left(\left. \dfrac{\partial J}{\partial x} \right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} \right)^2 & \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left(\left. \dfrac{\partial J}{\partial x} \dfrac{\partial J}{\partial y} \right)\right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} \\ \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left(\left. \dfrac{\partial J}{\partial x} \dfrac{\partial J}{\partial y} \right)\right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} & \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left(\left. \dfrac{\partial J}{\partial y} \right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} \right)^2 \end{pmatrix}^{-1} . \\ &\begin{pmatrix} \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left[I(\mathbf{y}) \left. \dfrac{\partial J}{\partial x} \right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} \right] \\ \sum\limits_{\mathbf{y} \in B_{\mathbf{x}}} \left[I(\mathbf{y}) \left. \dfrac{\partial J}{\partial y} \right|_{\mathbf{y}+\widehat{\mathbf{v}}_k(\mathbf{x})} \right] \end{pmatrix} \overset{\text{not}}{=} \widehat{\mathbf{v}}_k(\mathbf{x}) + \mathbf{A}(\mathbf{x}, \widehat{\mathbf{v}}_k(\mathbf{x})) \cdot \mathbf{b}(\mathbf{x}, \widehat{\mathbf{v}}_k(\mathbf{x})) \ , \ k \geq 0 \, . \end{aligned} \tag{2.9}$$

Les détails d'implantation de l'algorithme de Lucas et Kanade seront aussi présentés dans la section 2.2.1.4.

Remarquons que cette technique d'estimation au sens des moindres carrés, illustrée ici dans les cas d'un champ de déplacement localement translationnel, peut être généralisée sans difficulté à des modèles paramétriques de mouvement plus sophistiqués. Dans le cas d'un modèle

polynomial [Odobez95], $(u, v) = (P(x, y), Q(x, y))$, la minimisation effectuée dans l'espace des paramètres du modèle (les coefficients des polynômes P et Q) conduit à un estimateur linéaire de type (2.9), où les éléments de la matrice $\mathbf{A}$ et du vecteur $\mathbf{b}$ seront des polynômes d'ordre supérieur en $\frac{\partial J}{\partial x}$ et $\frac{\partial J}{\partial y}$.

Quénot [Quénot92] propose un algorithme original d'estimation du flot optique qui vise à faire décroître la complexité du calcul en transformant l'appariement par corrélation bidimensionelle en une série de procédures de corrélation unidimensionnelle. Explicitement, cet algorithme divise les deux images pour lesquelles on calcule le flot optique en bandes parallèles semi-recouvrantes et apparie de façon optimale des bandes de la première et de la deuxième image, en utilisant une technique de programmation dynamique. L'alignement est effectué par la minimisation d'une fonction de coût qui assure un recalage optimal entre des vecteurs colonne de chaque paire de bandes (approche unidimensionnelle). Avec les notations de la Figure 2.1, aligner d'une manière optimale les bandes k des deux images revient à établir un chemin optimal de recalage donnant la correspondance entre les vecteurs colonne des deux bandes. A partir d'un point d'initialisation (x, y), un chemin de recalage est construit progressivement à l'intérieur d'une zone (en gris clair) correspondant à un déplacement absolu maximal de m pixels, de façon à respecter une des trois directions de croissance suivante : $(x+1, y)$, $(x+1, y+1)$, $(x, y+1)$. Un coût est associé à chaque chemin de recalage possible et le chemin de recalage optimal sera défini comme étant celui de coût minimal.

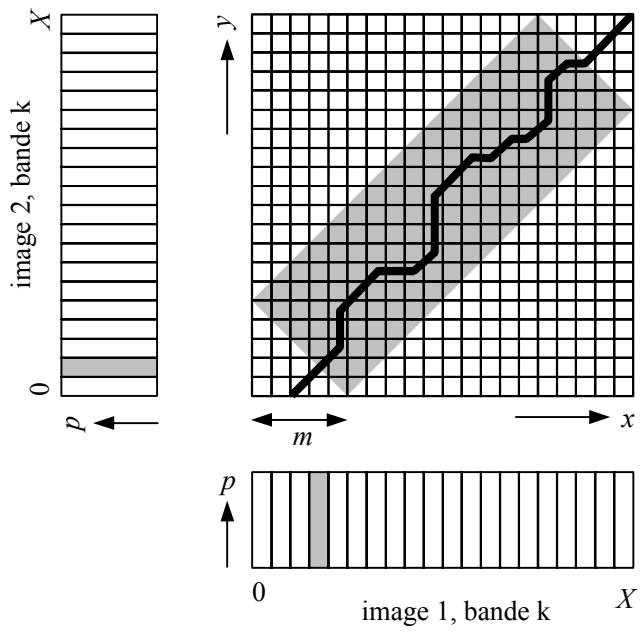


Figure 2.1 : Calcul du chemin optimal de recalage dans l'algorithme de Quénot [Quénot92].

La procédure de calcul du flot optique est effectuée itérativement dans une sorte de traitement hiérarchique comme suit : à chaque pas de l'algorithme, l'orientation des bandes parallèles change de 90° et leur largeur, ainsi que l'espacement inter-bande diminuent. En utilisant le champ de déplacements mis à jour au pas précédent, une image intermédiaire est calculée en déformant la deuxième image vers la première et ensuite utilisée en conjonction avec la première image pour le raffinement du champ de déplacements au pas suivant.

Cet algorithme, appliqué à des séquences de test utilisées par la communauté scientifique pour évaluer les performances des algorithmes de flot optique, conduit aux meilleurs résultats [Quénot96], ce qui a motivé notre intérêt pour cette approche.

2.2.1.3 Approches multirésolution

Les algorithmes différentiels de flot optique ainsi que l'algorithme de Lucas et Kanade présenté dans la section précédente utilisent des approximations dérivées de l'hypothèse de petits déplacements. Par conséquent, ces algorithmes estiment de manière fiable des déplacements ‹‹pixéliques››, cette contrainte limitant fortement leur application directe. Pour surmonter cet inconvénient, les stratégies d'estimation multirésolution (ou hiérarchique) [Enkelmann86, Glazer87] sont couramment mises en œuvre. L'idée de base consiste à utiliser une estimation incrémentale, distribuée sur plusieurs niveaux de résolution spatiale, de manière à satisfaire la condition des petits déplacements à chaque niveau.

La représentation multirésolution d'une image [Burt84] s'appuie sur la construction récursive d'une "pyramide" d'images dont chaque niveau est obtenu par un filtrage passe-bas (typiquement gaussien) du niveau précédent, suivi d'un sous-échantillonnage par un facteur 2 sur chaque axe de coordonnées, tandis que le premier niveau est donné par l'image même (Figure 2.2). Le filtrage passe-bas limite les effets de repliement spectral liés au sous-échantillonnage spatial.

Disposant d'une telle représentation, l'estimation multirésolution du flot optique commence au niveau correspondant à la résolution la plus basse, choisie telle que les déplacements restent ‹‹pixéliques››, et elle se propage de niveau en niveau, en se raffinant progressivement.

Pour formaliser l'extension multirésolution de l'algorithme de **Horn et Schunk** notons par $I^{(n)}(\mathbf{x}, t)$, $n = 0, 1, \ldots, N-1$, la pyramide associée à l'image $I(\mathbf{x}, t)$, avec $I^{(0)} = I$, et désignons par des indices supérieurs le niveau de la pyramide auquel les estimées sont associées. Supposant que l'estimation du champ de déplacements a été effectuée jusqu'au niveau $n+1$, soit $\widehat{\mathbf{v}}^{(n+1)}$, et prenant en compte que le passage au niveau inférieur de la pyramide implique la multiplication de la résolution par un facteur 2, la conservation de la luminance (2.3) au niveau

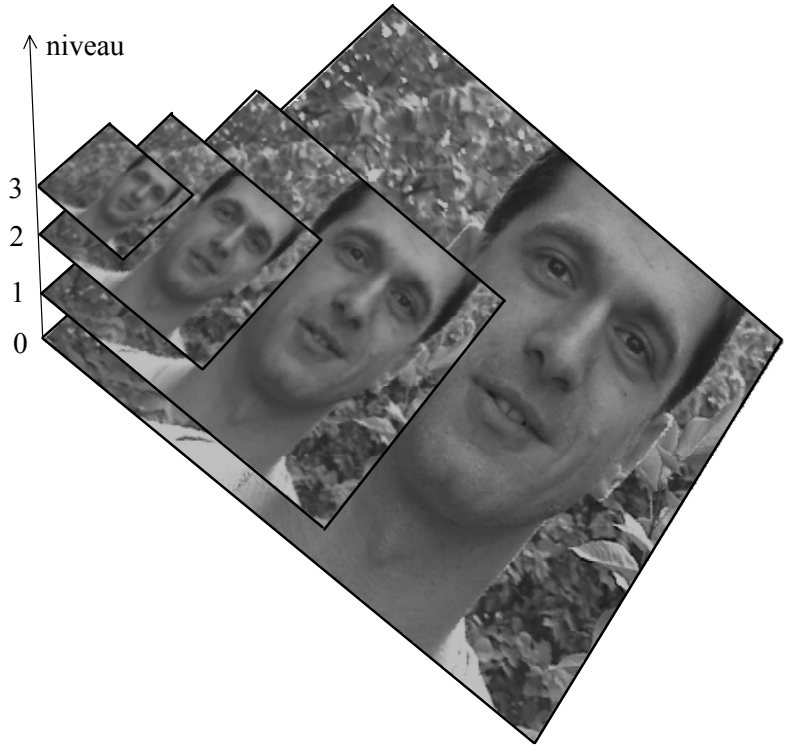


Figure 2.2 : Pyramide d'image à 4 niveaux obtenue récursivement par filtrage gaussien et sous-échantillonnage d'un facteur 2 sur chaque axe.

n dans le pixel $\mathbf{x}$ sera exprimée comme suit :

$$I^{(n)}(\mathbf{x} + 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2,\, t)\;\delta t + \delta\mathbf{v}^{(n)}(\mathbf{x}, t)\,\delta t\,,\; t + \delta t) = I^{(n)}(\mathbf{x} + 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2,\, t)\,\delta t\,,\; t)\;, \quad (2.10)$$

avec $\delta\mathbf{v}^{(n)}(\mathbf{x})$ à estimer, pour mettre à jour le champ de déplacement. Considérant à nouveau le développement de Taylor au premier ordre, (2.10) conduit à :

$$\nabla I^{(n)}(\mathbf{x} + 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2,\, t)\,\delta t\,,\; t) \cdot \delta\mathbf{v}^{(n)}(\mathbf{x},\, t) + I_t^{(n)}(\mathbf{x} + 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2,\, t)\,\delta t\,,\; t) = 0\;, \quad (2.11)$$

qui est une équation de la même forme que (2.2), et qui peut être résolue sous les contraintes de régularité de Horn et Schunk, en utilisant les itérations (2.6). Cela nécessite le calcul numérique de la dérivée I_t à tout niveau n. Notons que l'évaluation de la dérivée temporelle en terme de différences finies n'est valable que pour des petits déplacements, typiquement 2 pixels [Simoncelli99]. Au-delà de cette limite, la dérivée temporelle ne peut pas être estimée de manière précise, suite aux effets de repliement spectral liés au sous-échantillonnage temporel. Par conséquent, cette manière de résoudre l'estimation multirésolution du champ de vitesses,

ne peut pas conduire à des résultats robustes. La Figure 2.3 (c) montre le résultat du calcul du champ de vitesses entre les images présentées Figure 2.3 (a) et (b) (translation de 10 pixels), obtenu par la résolution directe de (2.11) dans un schéma pyramidal à 4 niveaux.

Pour surmonter la limitation liée au calcul de la dérivée temporelle, la solution [Enkelmann86] consiste à calculer, à chaque niveau n, une image intermédiaire $I'^{(n)}$, obtenue par la déformation de $I^{(n)}$ en fonction du champ de vitesses estimé au niveau $n+1$:

$$I'^{(n)}(\mathbf{x},\, t) = I^{(n)}(\mathbf{x} - 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2,\, t)\,\delta t\,,\, t) \tag{2.12}$$

et à appliquer sur celle-ci l'algorithme monorésolution d'estimation du champ de vitesses. La figure 2.3 (d) présente le résultat de l'estimation correcte du champ de vitesses correspondant à la translation de 10 pixels.

En conclusion, l'algorithme de Horn et Schunk en version multirésolution sera le suivant :

1. calculer la pyramide d'images $I^{(n)}\,(\mathbf{x},\, t)$, $n = 0,\, 1,\, \ldots,\, N-1$;
2. $n = N-1$; $\widehat{\mathbf{v}}^{(N)}\,(\mathbf{x},\, t) = 0$, pour tout $\mathbf{x} \in \operatorname{supp} I^{(N)}$;
3. calculer $I'^{(n)}$ avec la formule (2.12) et ensuite calculer les dérivées $\nabla I'^{(n)}$ et $I_t'^{(n)}$;
4. $\widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x}, t) = 0$, pour tout $\mathbf{x} \in \operatorname{supp} I'^{(n)}$;
5. pour tout $\mathbf{x} \in \operatorname{supp} I'^{(n)}$ calculer $\overline{\widehat{\delta\mathbf{v}}}^{(n)}\,(\mathbf{x},\, t)$ avec la formule (2.4) et ensuite :

$$\widehat{\delta\mathbf{v}}_{\text{new}}^{(n)}(\mathbf{x},\, t) = \overline{\widehat{\delta\mathbf{v}}}^{(n)}\,(\mathbf{x},\, t) - \nabla I'^{(n)}(\mathbf{x},\, t)\frac{\nabla I'^{(n)}(\mathbf{x},\, t)\cdot\overline{\widehat{\delta\mathbf{v}}}^{(n)}\,(\mathbf{x},\, t) + I_t'^{(n)}(\mathbf{x},\, t)}{\alpha^2 + ||\nabla I'^{(n)}(\mathbf{x},\, t)||^2}\ .$$

si $\max\limits_{\mathbf{x}} ||\widehat{\delta\mathbf{v}}_{\text{new}}^{(n)}(\mathbf{x},\, t) - \widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x},\, t)|| > \varepsilon\,(n)$

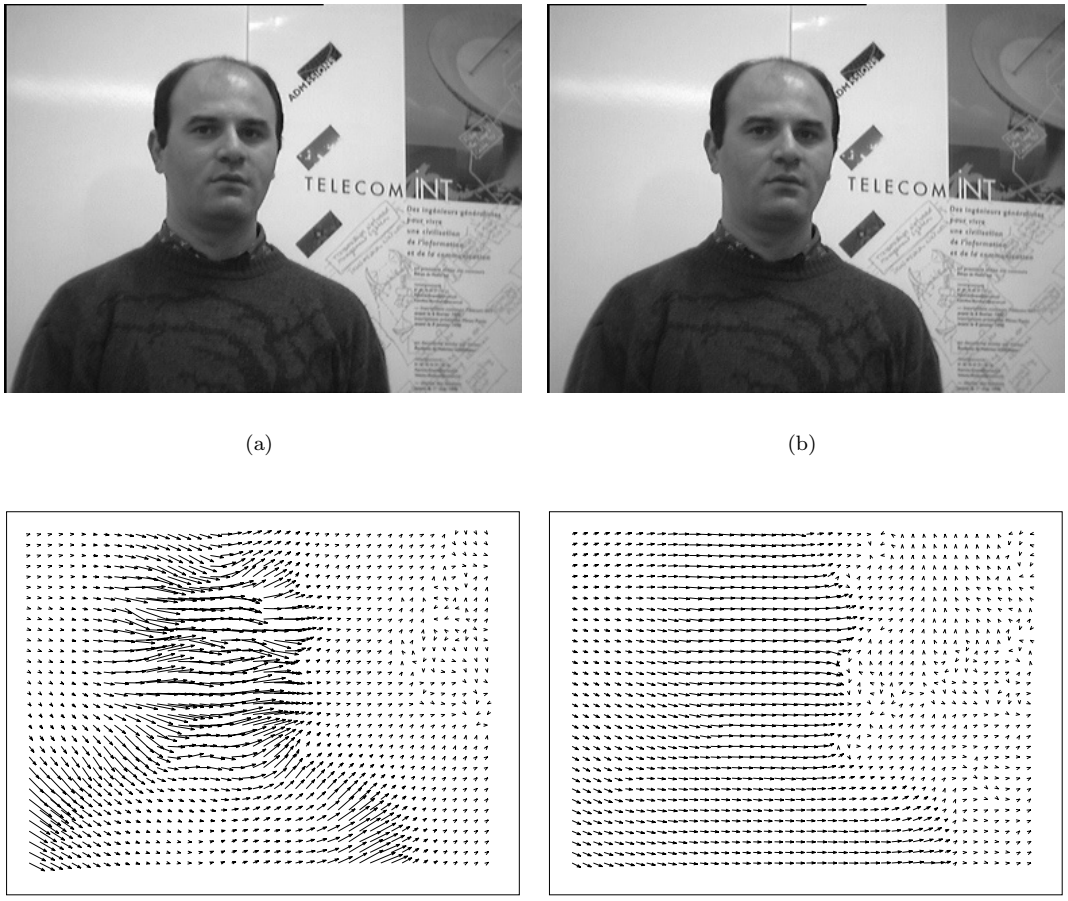
$\widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x},\, t) = \widehat{\delta\mathbf{v}}_{\text{new}}^{(n)}(\mathbf{x},\, t)$, pour tout $\mathbf{x} \in \operatorname{supp} I'^{(n)}$;
aller à 3 ;

sinon

$\widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x},\, t) = \widehat{\delta\mathbf{v}}_{\text{new}}^{(n)}(\mathbf{x},\, t)$, pour tout $\mathbf{x} \in \operatorname{supp} I'^{(n)}$;

6. pour tout $\mathbf{x} \in \operatorname{supp} I'^{(n)}$ calculer $\widehat{\mathbf{v}}^{(n)}\,(\mathbf{x},\, t) = 2\,\widehat{\mathbf{v}}^{(n+1)}\,(\mathbf{x},\, t) + \widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x},\, t)$;
si $n > 0$

$n := n-1$;
aller à 2 ;



(a) (b)

(c) (d)

Figure 2.3 : Estimation du champ de vitesses par l'algorithme de Horn et Schunk en version multirésolution : (a) et (b) images de test (translation de 10 pixels) ; (c) flot optique erroné, obtenu sans prendre en compte les effets de repliement spectral liés au sous-échantillonnage temporel ; (d) flot optique correctement estimé.

sinon

retourner $\widehat{\mathbf{v}}^{(n)}$;
ARRET ;

où $\varepsilon\,(n)$ est un seuil, dépendant du niveau n, qui contrôle le critère d'arrêt des itérations de Horn et Schunk.

L'extension multirésolution de la méthode de **Lucas et Kanade** est immédiate. Soient I et J deux images données. L'algorithme permettant d'estimer le vecteur de déplacement $\widehat{\mathbf{v}}(\mathbf{x})$ au pixel $\mathbf{x}$ sera :

1. calculer les pyramides d'images $I^{(n)}$ et $J^{(n)}$, $n = 0, 1, \ldots, N-1$;

2. $n = N - 1$; $\mathbf{x} := \mathbf{x}/2^{N-1}$; $\widehat{\mathbf{v}}^{(N)}(\mathbf{x}/2)$;
3. $\widehat{\mathbf{v}}^{(n)}(\mathbf{x}) = 2\,\widehat{\mathbf{v}}^{(n+1)}(\mathbf{x}/2)$;
4. calculer

   $\widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x}) = \mathbf{A}(\mathbf{x}, \widehat{\mathbf{v}}^{(n)}(\mathbf{x})) \cdot \mathbf{b}(\mathbf{x}, \widehat{\mathbf{v}}^{(n)}(\mathbf{x}))$, avec $\mathbf{A}$ et $\mathbf{b}$ donnés par la formule (2.9) appliquée pour $I^{(n)}$ et $J^{(n)}$;

   $\widehat{\mathbf{v}}^{(n)}(\mathbf{x}) := \widehat{\mathbf{v}}^{(n)}(\mathbf{x}) + \widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x})$;

   si $||\widehat{\delta\mathbf{v}}^{(n)}(\mathbf{x})|| \leq \varepsilon(n)$

   aller à 3 ;
5. *si* $n > 0$

   $\mathbf{x} := 2\,\mathbf{x}$;
   $n := n - 1$;
   aller à 2 ;

   sinon

   retourner $\widehat{\mathbf{v}}^{(n)}(\mathbf{x})$;
   ARRET ;

où $\varepsilon(n)$ contrôle le critère d'arrêt des itérations de Lucas et Kanade au niveau n.

L'algorithme de Quénot intègre *a priori* un schéma de traitement hiérarchique lui permettant d'estimer, en mode monorésolution et de manière fiable, les vecteurs de déplacement ayant une amplitude maximum de 10% de la taille de l'image, équivalent aux performances des algorithmes multirésolution.

2.2.1.4 Implantation algorithmique

Les algorithmes de Horn et Schunk et de Lucas-Kanade nécessitent le calcul des dérivées de l'image, $\dfrac{\partial I}{\partial x}$ et $\dfrac{\partial I}{\partial y}$. Dans notre implantation, celles-ci sont évaluées à l'aide de l'algorithme de Canny [Canny86], *i.e.* filtrage linéaire monodimensionnel dans les deux directions x et y avec un noyau correspondant à la dérivée d'une gaussienne dont le paramètre σ est fixé à 1.0, ce qui nous conduit à limiter le support du filtre à 5 pixels. La dérivée temporelle qui intervient dans l'algorithme de Horn et Schunk est évaluée comme la différence finie au premier ordre entre deux images successives. Cela limite à deux trames le support temporel sur lequel l'estimation multirésolution est effectuée. En outre, si le calcul des valeurs inter-pixels s'impose, celui-ci s'effectue à l'aide d'une interpolation bilinéaire.

Dans les approches multirésolution, les pyramides d'images résultent d'un filtrage gaussien dont le paramètre de lissage σ vaut 1.5 et d'un sous-échantillonnage d'un facteur 2 dans les deux directions. Le support de la fenêtre du filtre gaussien est limité à 5×5 pixels, cela assurant ainsi un bon compromis entre le volume de calcul et la qualité des résultats.

Le paramètre de lissage α dans l'algorithme de Horn et Schunk est fixé expérimentalement à 20, cette valeur conduisant à de meilleurs résultats que la valeur 100, suggérée par les auteurs.

2.2.1.5 Résultats : analyse qualitative et quantitative

Les Figures 2.4 (c) et (d) présentent les champs de déplacements résultant de l'application de l'algorithme du Horn et Schunk aux images de test présentées Figures 2.4 (a) et (b) (mouvement global dominant combiné avec des déformations locales et des occultations) pour deux valeurs du paramètre de lissage, respectivement $\alpha = 100$ et $\alpha = 20$. Une valeur élevée de α conduit à une diffusion importante du champ de déplacements à travers les contours occultés ainsi qu'à une faible représentation des mouvements locaux (la région correspondant à la bouche). Ces deux inconvénients sont éliminés dans le deuxième cas.

L'algorithme de Lucas et Kanade (Figure 2.4 (e)) enlève complètement la diffusion du flot optique entre les régions se caractérisant par des mouvements de type différent, car il n'impose pas de contraintes de régularité locale. Par conséquent, des erreurs locales importantes peuvent apparaître dans le voisinage des contours occultés.

La méthode de Quénot (Figure 2.4 (e)) fournit un flot optique globalement similaire à celui de l'algorithme de Horn et Schunk pour des valeurs faibles du paramètre α. Toutefois, remarquons un meilleur comportement en cas d'occultations, notamment sur les contours intérieurs des lèvres.

Pour évaluer objectivement les performances des algorithmes de flot optique étudiés, nous avons effectué des simulations numériques sur une séquence calibrée comportant 100 images correspondant à l'animation d'un maillage 3D de tête avec placage de texture, projeté sur un fond mobile (Figure 2.12). Les vecteurs de déplacement 2D sont connus dans chaque pixel de la projection ainsi obtenue ($\approx 10^6$ échantillons pour toute la séquence). L'amplitude inter-trame du mouvement 2D ovservé dépasse 20 pixels, ce qui nous à conduit à mettre en œuvre une représentation pyramidale à 4 niveaux.

Pour évaluer l'écart entre les vecteurs de déplacement réels $\mathbf{v}_{\text{real}}$ et les vecteurs estimés $\widehat{\mathbf{v}}$ nous utilisons une mesure comportant une composante d'erreur en amplitude :

$$\varepsilon_{\text{a}}(\mathbf{v}_{\text{real}}, \widehat{\mathbf{v}}) = \|\mathbf{v}_{\text{real}} - \widehat{\mathbf{v}}\| \ ,$$

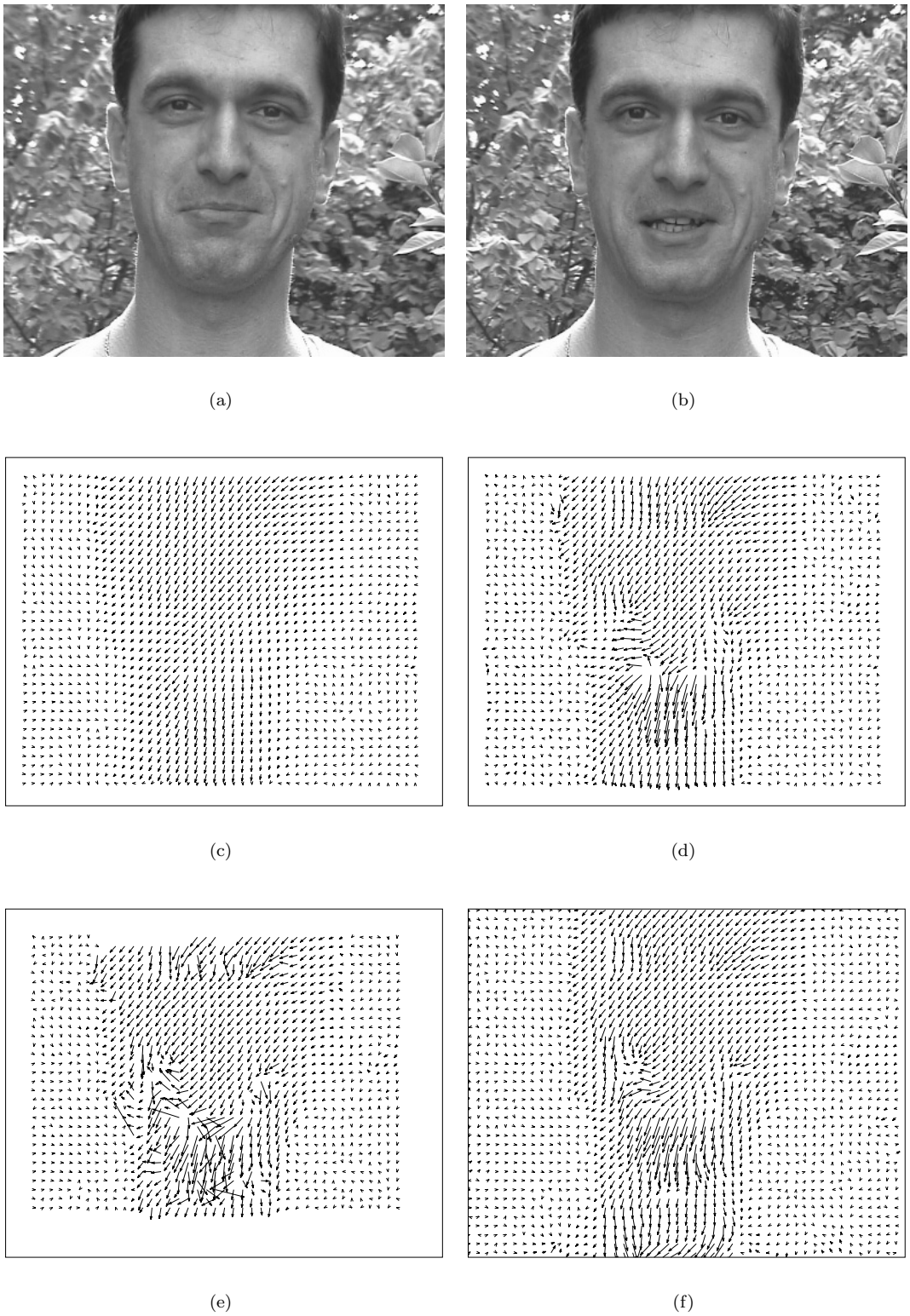


Figure 2.4 : Différentes estimations du flot optique : (a) et (b) images de test ; (c) et (d) algorithme de Horn et Schunk avec le paramètre de lissage $\alpha = 100$ et $\alpha = 20$, respectivement ; (e) algorithme de Lucas et Kanade ; (f) algorithme de Quénot.

$||\cdot||$ étant la norme L_2 et une composante d'erreur angulaire. Cette dernière est définie par l'angle formé entre les vecteurs spatio-temporaux $(\mathbf{v}_{\text{real}}, 1)$ et $(\widehat{\mathbf{v}}, 1)$ [Fleet90] :

$$\varepsilon_\phi(\mathbf{v}_{\text{real}},\, \widehat{\mathbf{v}}) = \measuredangle((\mathbf{v}_{\text{real}},\, 1),\, (\widehat{\mathbf{v}},\, 1)) = \arccos\left(\frac{(\mathbf{v}_{\text{real}},\, 1)}{\|(\mathbf{v}_{\text{real}},\, 1)\|} \cdot \frac{(\widehat{\mathbf{v}},\, 1)}{\|(\widehat{\mathbf{v}},\, 1)\|}\right) .$$

Les algorithmes de Horn et Schunk ($\alpha = 100$ et $\alpha = 20$), de Lucas et Kanade et de Quénot ont été successivement appliqués sur la séquence calibrée et les résultats ont été interprétés en termes de distributions statistiques des erreurs en amplitude ε_{a} et en angle ε_ϕ. La Figure 2.5 présente les fonctions de répartition déterminées expérimentalement pour chaque algorithme. En outre, les valeurs moyennes et les écarts types de ε_{a} et ε_ϕ sont indiqués dans le Tableau 2.1. De cette analyse, l'algorithme de Quénot apparaîtrait globalement le plus performant parmi les algorithmes présentés.

Algorithme	$\overline{\varepsilon_{\text{a}}}$ [pixels]	$\sqrt{\overline{\varepsilon_{\text{a}}^2} - \overline{\varepsilon_{\text{a}}}^2}$ [pixels]	$\overline{\varepsilon_\phi}$ [degrés]	$\sqrt{\overline{\varepsilon_\phi^2} - \overline{\varepsilon_\phi}^2}$ [degrés]
Horn et Schunk ($\alpha = 100$)	1.43	1.16	12.45	12.39
Horn et Schunk ($\alpha = 20$)	0.74	1.25	4.36	11.78
Lucas et Kanade	0.72	1.80	8.46	17.85
Quénot	0.63	0.93	7.46	11.17

Tableau 2.1 : Valeurs moyennes et écarts types des erreurs d'estimation du champ de déplacements par différents algorithmes.

2.2.2 La méthode d'optimisation du simplexe

La méthode du simplexe, proposée par Nelder et Mead [Nelder65] est une technique performante d'optimisation numérique, reconnue par sa robustesse [Press98] et de faible coût de calcul. L'algorithme du simplexe utilise uniquement les valeurs de la fonction-objectif. Eviter d'utiliser les dérivées renforce la robustesse par rapport au bruit. De plus, la méthode est indépendante de la dimension de l'espace sur lequel la fonction-objectif est définie.

Un simplexe de $\mathbf{R}^n$ est un polyèdre se composant de $n+1$ sommets et de tous les segments reliant ces sommets (un simplexe est un segment dans $\mathbf{R}$, un triangle dans $\mathbf{R}^2$ et un tétraèdre dans $\mathbf{R}^3$). Un simplexe n-dimensionnel est dit non dégénéré si chacun de ses sommets forme avec les n autres sommets un ensemble de vecteurs linéaires indépendants.

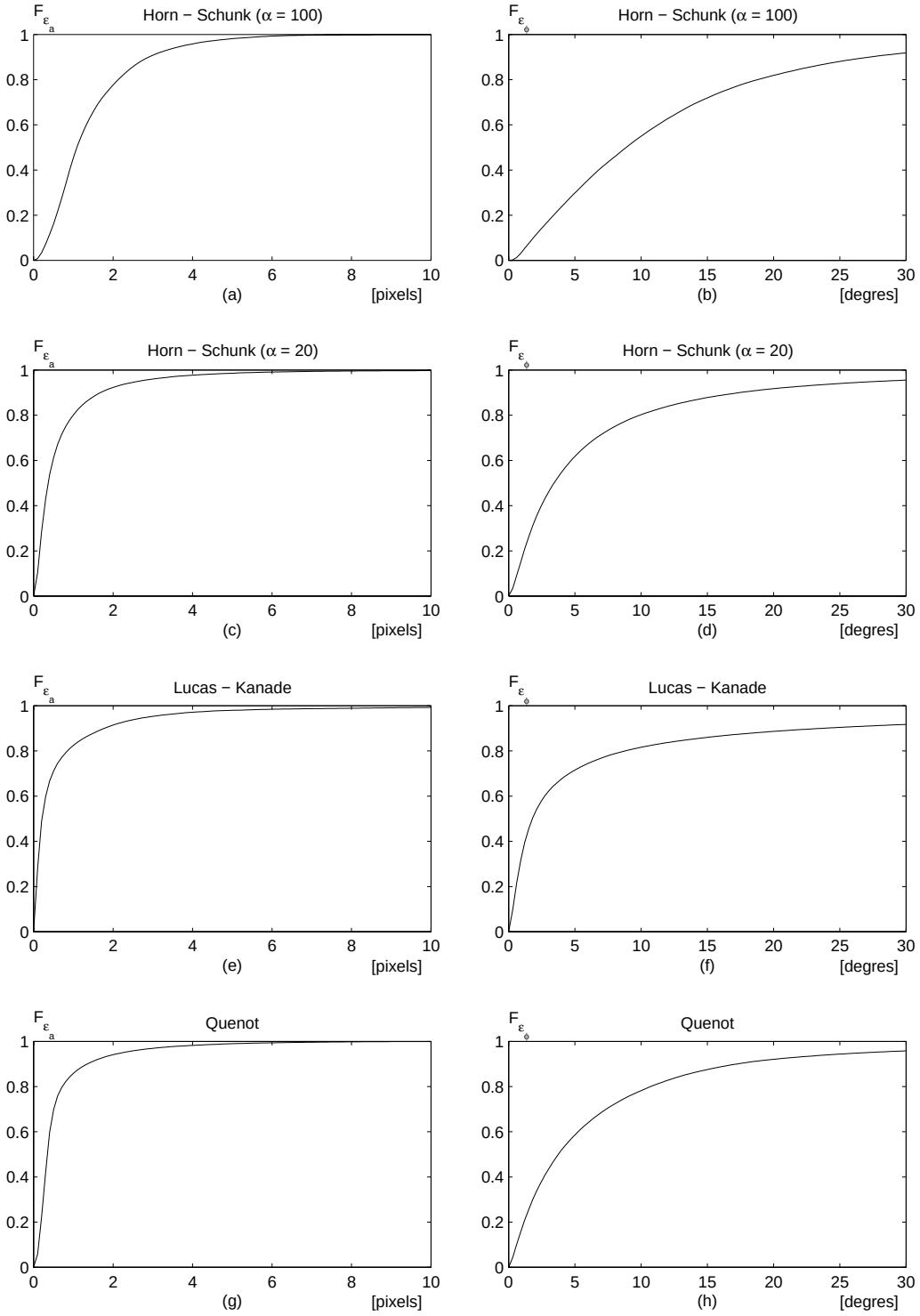


Figure 2.5 : Fonctions de répartition des erreurs d'estimation du champ de déplacements correspondant aux algorithmes : (a), (b) et (c), (d) Horn et Schunk avec pour paramètre de lissage $\alpha = 100$ et $\alpha = 20$, respectivement ; (e), (f) Lucas et Kanade ; (g), (h) Quénot. ε_{a} et ε_{ϕ} désignent respectivement l'erreur en amplitude et en angle.

La minimisation du simplexe appliquée à une fonction réelle $f : \mathbf{R}^n \to \mathbf{R}$ consiste à mettre à jour itérativement un simplexe n-dimensionnel tel que les valeurs de la fonction calculées aux sommets décroissent. Pour détailler cet algorithme, utilisons les notations suivantes :

- $\mathbf{s}_i$, $i = 0, 1, \ldots, n$ désignent les sommets d'un simplexe n-dimensionnel non dégénéré ;
- $\overline{\mathbf{s}} = \dfrac{1}{n+1} \displaystyle\sum_{i=0}^{n} \mathbf{s}_i$, est le centre de gravité du simplexe ;
- $f_i = f(\mathbf{s}_i)$, $i = 0, 1, \ldots, n$ sont les valeurs de la fonction-objectif calculées aux sommets du simplexe ;
- $l = \arg\min\limits_i \ f_i$ et $l = \arg\max\limits_i \ f_i$;
- $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_3$, et $\mathcal{I}_4$, les intervalles réels définis comme suit :

$$\mathcal{I}_1 = (-\infty,\ f_l] \ , \qquad \mathcal{I}_2 = \{x \mid f_l < x \leq \max_{i \neq h} f_i\} \ ,$$
$$\mathcal{I}_3 = \{x \mid \max_{i \neq h} f_i < x \leq f_h\} \ , \qquad \mathcal{I}_4 = (f_h,\ \infty) \ .$$

L'algorithme de minimisation consiste à enchaîner les cinq opérations géométriques de base suivantes :

Initialisation :

définir un simplexe n-dimensionnel initial non dégénéré ; la procédure la plus triviale consiste à prendre un point $\mathbf{s}_0$ et à définir les n autres points par :

$$\mathbf{s}_i = \mathbf{s}_0 + \lambda_i \, \mathbf{e}_i \ , \ i = 1, 2, \ldots, n \ ,$$

où $\mathbf{e}_i$ désignent les vecteurs unité des axes des coordonnées dans $\mathbf{R}^n$ et λ_i sont des paramètres reflétant l'échelle de variation des variables de la fonction-objectif,

Réflexion :

définir le point de réflexion $\mathbf{s}'$ comme suit :

$$\mathbf{s}' = \overline{\mathbf{s}} + \alpha \, (\overline{\mathbf{s}} - \mathbf{s}_h) \ , \text{ avec } \alpha > 0 \ ,$$

Expansion :

définir le point d'expansion $\mathbf{s}''$ comme suit :

$$\mathbf{s}'' = \overline{\mathbf{s}} + \beta \, (\mathbf{s}' - \overline{\mathbf{s}}) \ , \text{ avec } \beta > 1 \ ,$$

Contraction :

définir le point de contraction $\mathbf{s}'''$ comme suit :

$$\mathbf{s}''' = \overline{\mathbf{s}} + \gamma\,(\mathbf{s}_h - \overline{\mathbf{s}}) \text{ , avec } 0 < \gamma < 1 \text{ ,}$$

Rétrécissement :

remplacer chaque sommet $\mathbf{s}_i$ du simplexe par $(\mathbf{s}_i + \mathbf{s}_l)/2$.

Ces opérations dans le cas bidimensionnel sont illustrées Figure 2.6. L'algorithme se déroule de la manière suivante :

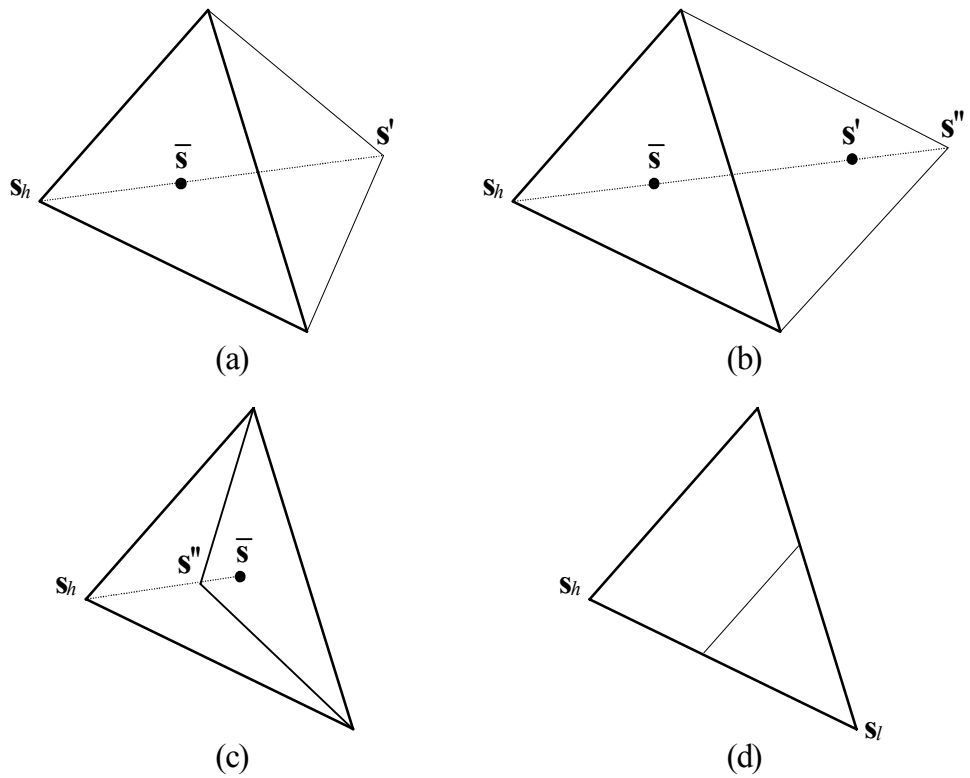


Figure 2.6 : Les opérations géométriques de base de l'algorithme du simplexe : (a) réflexion, (b) expansion, (c) contraction, (d) rétrécissement.

INITIALISATION : effectuer *initialisation* ;

REFLEXION : effectuer *réflexion* ;

si $f(\mathbf{s}') \in \mathcal{I}_1$, *aller à* EXPANSION ;

si $f(\mathbf{s}') \in \mathcal{I}_2$, remplacer $\mathbf{s}_h$ par $\mathbf{s}'$ et *aller à* FIN DE CYCLE ;

si $f(\mathbf{s}') \in \mathcal{I}_3$, remplacer $\mathbf{s}_h$ par $\mathbf{s}'$ et *aller à* CONTRACTION ;

si $f(\mathbf{s}') \in \mathcal{I}_4$, *aller à* CONTRACTION ;

EXPANSION : effectuer *expansion* ;

si $f(\mathbf{s}'') \in \mathcal{I}_1$ remplacer $\mathbf{s}_h$ par $\mathbf{s}''$ et *aller à* FIN DE CYCLE ;

sinon remplacer $\mathbf{s}_h$ par $\mathbf{s}'$ et *aller à* FIN DE CYCLE ;

CONTRACTION : effectuer *contraction* ;

si $f(\mathbf{s}''') \in \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$ remplacer $\mathbf{s}_h$ par $\mathbf{s}'''$ et *aller à* FIN DE CYCLE ;

sinon aller à RETRECISSEMENT ;

RETRECISSEMENT : effectuer *rétrécissement* et *aller à* FIN DE CYCLE ;

FIN DE CYCLE :

si le critère d'arrêt est satisfait *retourner* $\mathbf{s}_l$; ARRET ;

sinon aller à REFLEXION ;

Le critère d'arrêt est un test usuel utilisé dans les méthodes d'optimisation numérique (la variation de l'estimée $\mathbf{s}_l$ ou de la valeur de la fonction-objectif $f(\mathbf{s}_l)$ entre deux itérations successives est inférieure à un certain seuil).

Observations :

1. chaque itération nécessite au moins deux évaluations de la fonction-objectif ;
2. les valeurs des paramètres intervenant dans les opérations géométriques de l'algorithme sont empiriques. Elles doivent être adaptées au problème d'optimisation. Des valeurs raisonnables de départ sont celles suggérées par Nelder et Mead, notamment $\alpha = 1$, $\beta = 2$ et $\gamma = 1/2$;
3. pour diminuer les risques de blocage dans des minima locaux, la procédure du simplexe peut être répétée systématiquement. Dans ce cas, le point de minimum détecté servira pour réinitialiser l'algorithme.

2.3 Modélisation par flot coloré

L'estimation de la pose 3D de la tête proposée ici s'appuie sur la donnée d'un modèle 3D générique de tête, un choix de primitives 2D d'image et 3D du modèle puis sur une mise en correspondance par la méthode du simplexe. Dans la suite, nous détaillons successivement chacune de ces étapes.

2.3.1 Modélisation 3D de la tête

Dans le contexte de notre étude, nous proposons de construire un modèle 3D de tête, de complexité géométrique *ad hoc*, afin de garantir un compromis satisfaisant entre robustesse de l'estimation et volume de calcul. Nous avons le choix entre deux approches possibles de synthèse : 1) par maillages polygonaux et 2) par surfaces analytiques.

Les modèles maillés offrent l'avantage d'une modélisation fidèle des détails du visage (yeux, nez, bouche, sourcils) permettant ainsi une simulation réaliste de leurs déformations. Elles présentent toutefois deux inconvénients majeurs : la difficulté d'acquisition et la complexité élevée. En effet, ce type de modèle nécessite des techniques d'acquisition sophistiquées (laser, par exemple) ou doit être emprunté à des bases d'objets virtuels. Les maillages réalistes de tête (Figure 1.2 (a)) comportent des milliers de sommets, d'arêtes et de facettes à la géométrie non nécessairement triangulaire, ne vérifiant pas de propriétés de régularité (par exemple, absence de sommet en T) et pouvant présenter une topologie complexe (pas de simple connexité, en général). D'un coût de stockage important, ils nécessitent de recourir à des algorithmes de codage monorésolution [Curila99] ou multirésolution [Taubin98]. Ces derniers présentent l'avantage de fournir des représentations maillées à un niveau de simplicité susceptible de nous intéresser, mais ne garantissent pas une distribution morphologiquement satisfaisante des facettes. En outre, le caractère générique du modèle est hypothéqué même si les aspects déformation peuvent être pris en compte [Huang93].

En ce qui concerne le deuxième type d'approche, une description analytique conduit à une représentation compacte du modèle et à une manipulation très aisée des données dans un contexte déformable ou non. De plus, pour les surfaces analytiques, la détection des parties visibles par rapport à un point d'observation nécessite un volume de calcul moins important que dans le cas des maillages polygonaux. Ce sont les principales raisons qui nous ont conduit à opter pour une synthèse analytique du modèle 3D de tête.

Un modèle analytique simple de tête a été utilisé par Basu *et al.* [Basu96] sous forme d'un ellipsoïde de rotation. Toutefois, l'ellipsoïde ne s'adapte pas très bien à la morphologie de la tête, constituant ainsi l'une des principales sources d'erreur dans l'estimation de la pose 3D. Un modèle 3D plus adapté a été proposé par Tarel et al [Tarel98] qui approchent la tête par une surface algébrique fermée, obtenue suite à un ajustement polynomial sur des données 3D. Cette méthode conduit à de bons résultats, mais elle est peu pratique puisque le degré de la surface devient important pour des approximations satisfaisantes. De plus, cette approche nécessite une acquisition 3D.

Pour s'affranchir de ces inconvénients, nous avons développé une méthode de génération, à

partir de données 2D, d'un modèle de type surface analytique spécifiant la géométrie globale de la tête en négligeant les caractères morphologiques secondaires. Le modèle est obtenu par ajustement sur un ensemble de points de contour d'une tête quelconque (d'un adulte ou d'un enfant) considérée sous trois incidences, d'une surface exprimée sous forme d'une série de Fourier tronquée :

$$\begin{aligned} r(\theta,\, \varphi) \quad = \quad & \sum_{k,l=0}^{N} [A_{kl}\, \cos(k\,\theta)\cos(l\,\varphi) + B_{kl}\, \cos(k\,\theta)\sin(l\,\varphi) + \\ & C_{kl}\, \sin(k\,\theta)\cos(l\,\varphi) + D_{kl}\, \sin(k\,\theta)\sin(l\,\varphi)] \ , \end{aligned} \tag{2.13}$$

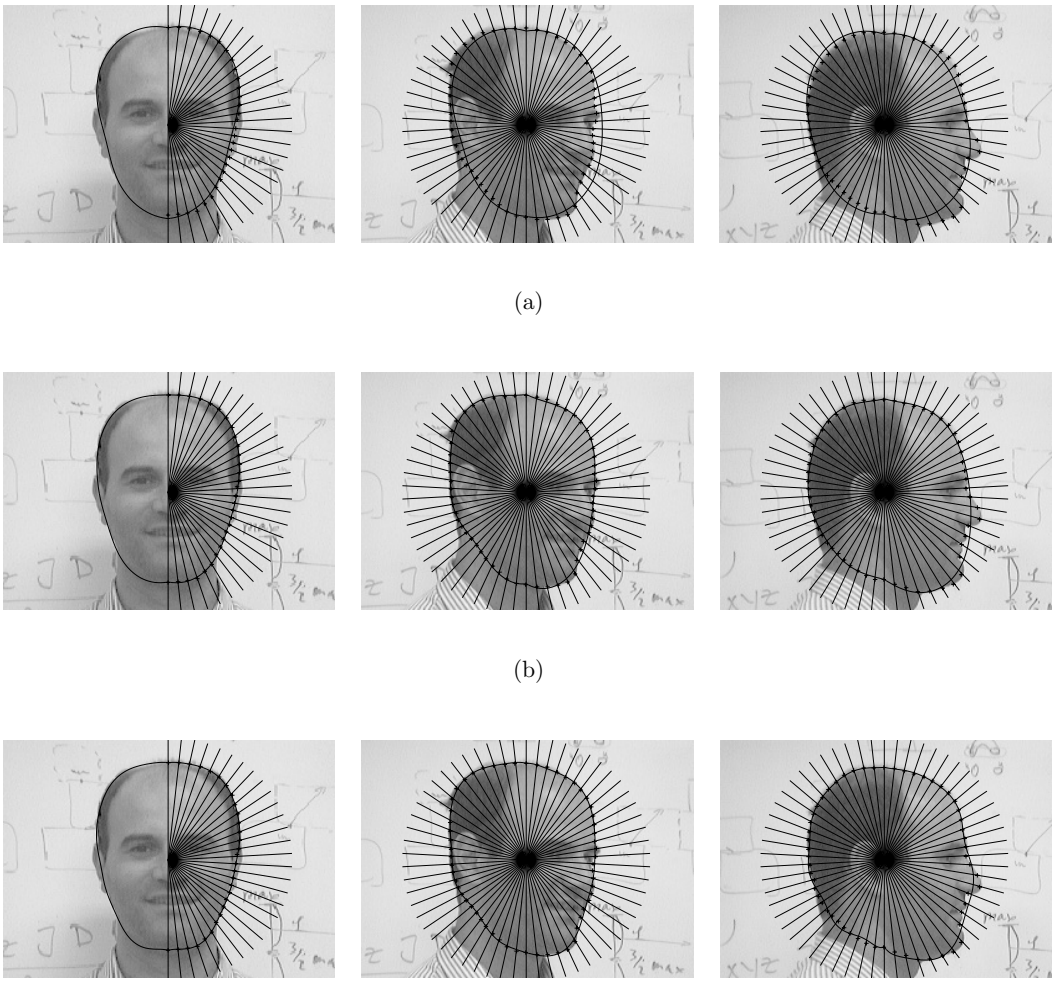
où θ et φ sont les coordonnées curvilignes sur la surface du modèle (coordonnées sphériques) et r la distance de l'origine aux points de la surface.

Les coefficients de la série sont calculés comme suit : considérant trois images de tête prises sous trois incidences distinctes, respectivement de face, semi-profil, et profil (Figure 2.7), nous sélectionnons interactivement un ensemble de points sur le contour de la tête pour lesquels nous mesurons r, θ et φ. Particularisant l'équation (2.13) pour chaque triplet (r, θ, φ), nous obtenons un système linéaire d'équations, dont les inconnues sont A_{kl}, B_{kl}, C_{kl}, D_{kl}. Pour un nombre de points suffisamment grand, ce système est surdéterminé et les coefficients A_{kl}, B_{kl}, C_{kl}, D_{kl} sont calculés comme solution au sens généralisé (obtenue par la méthode de décomposition en valeurs singulières [Press98]). Si θ et φ désignent respectivement les angles d'azimut et d'élévation en coordonnées sphériques, pour obtenir une surface symétrique par rapport à un plan vertical perpendiculaire sur le visage, les coefficients C_{kl} et D_{kl} sont forcés à zéro. En outre, pour que la surface soit fermée, deux contraintes supplémentaires doivent être ajoutées :

$A_{kl} = 0$, si k et l sont impairs, ou si l est pair,
$B_{kl} = 0$, si k est impair et l est pair, ou si l est impair.

La précision de l'approximation est liée à l'ordre de la série et au nombre de points d'échantillonnage considérés. La Figure 2.7 montre l'ajustement des points de contour de la tête sous les trois incidences considérées, pour $N = 2$, 4 et 6, respectivement (*i.e.* les séries comportent 15, 40 et 77 coefficients, conformément aux contraintes imposées).

La Figure 2.8 présente les surfaces ainsi obtenues. Le modèle généré par la série trigonométrique d'ordre $N = 4$ (Figure 2.8 (b)) offre un bon compromis entre la précision de la modélisation et la compacité de la représentation ; c'est la raison pour laquelle nous l'avons choisi dans nos expérimentations. Disposant de ce modèle de tête, la méthode d'estimation de la pose consiste en une mise en correspondance des primitives 3D du modèle avec les primitives 2D d'image.



(a)

(b)

(c)

Figure 2.7 : Pour trois incidences : Ajustement des points de contour de la tête par une série trigonométrique d'ordre : (a) $N = 2$; (b) $N = 4$; (c) $N = 6$.

2.3.2 Primitives 2D de l'image et 3D du modèle

Le premier type de primitives d'image correspond à des informations de mouvement dans la région du visage. L'information de mouvement peut être appréhendée à partir du champ de déplacements que différentes approches permettent d'estimer par l'intermédiaire du flot optique.

Comme nous l'avons précisé dans la section 2.2.1, les algorithmes différentiels de flot optique dont le prototype est celui de Horn et Schunk, ainsi que les approches par corrélation utilisant des formulations différentielles (Lucas et Kanade), estiment de manière fiable des déplacements ‹‹pixéliques››. Cependant, intégrées dans un schéma multirésolution d'ordre k, ces techniques peuvent prendre en compte des mouvements globaux pouvant aller jusqu'à k pixels. Notons

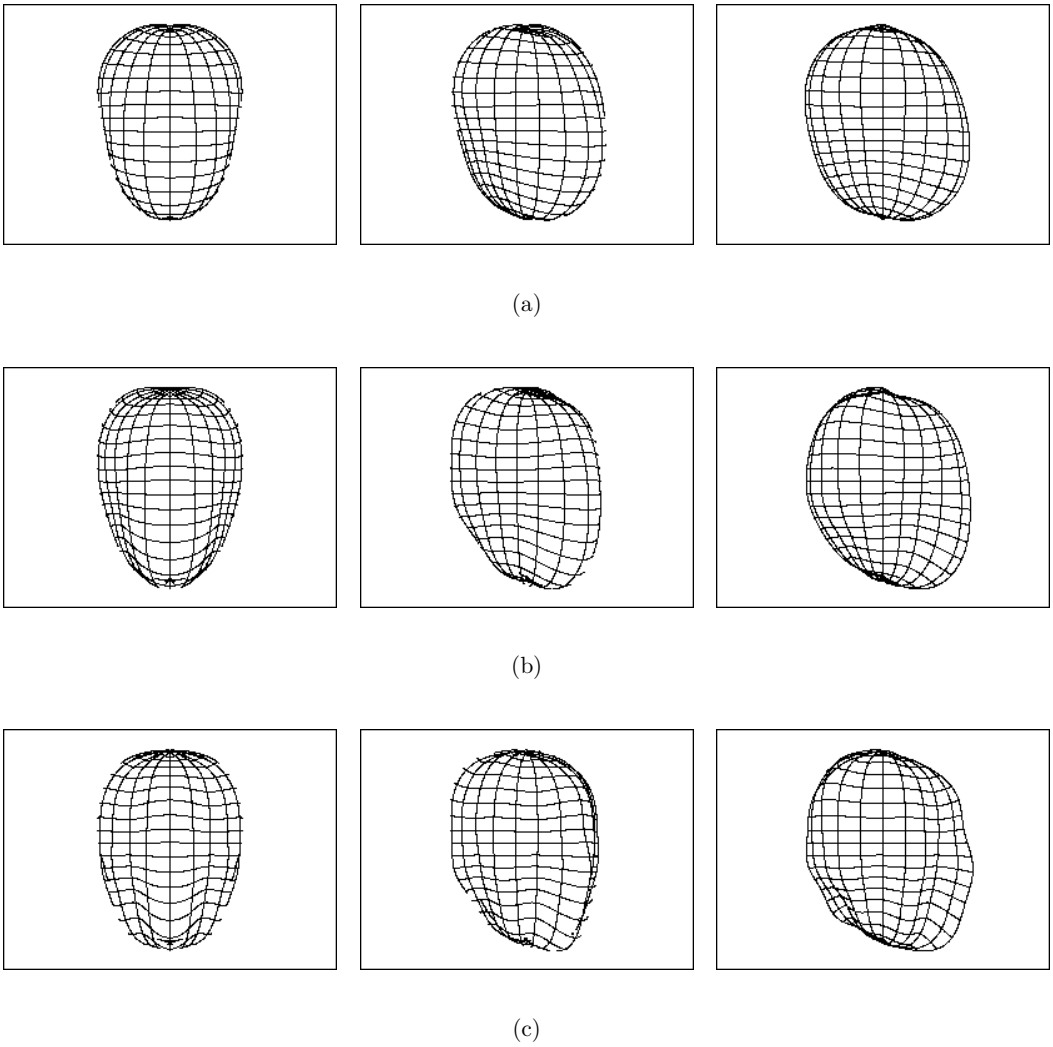


Figure 2.8 : Surfaces synthétisées correspondant aux différents ordres N de la série trigonométrique : (a) $N = 2$; (b) $N = 4$; (c) $N = 6$.

toutefois que les erreurs d'estimation se cumulent plus ou moins le long de la pyramide multirésolution. En outre, le recours à la multirésolution est limité par la taille des objets dont on cherche à estimer le mouvement. Dans le cas du visage, seule une pyramide à quatre niveaux est pertinente, ce qui permet d'estimer de façon fiable un mouvement d'au plus une vingtaine de pixels. Dans le cadre des approches de type appariement par blocs, la programmation dynamique orthogonale (Quénot) fournit des estimations robustes pour des déplacements allant jusqu'à une vingtaine de pixels (pour des images de taille usuelle), tout en se plaçant dans un contexte monorésolution. L'analyse comparative de ces méthodes en termes de précision, amplitude de mouvement pris en compte, rapidité de calcul, nous a conduit à retenir la méthode d'estimation du flot optique par programmation dynamique. La primitive du modèle 3D analogue au flot

optique est constituée par le champ de déplacements 3D théorique résultant du mouvement rigide du modèle.

Le second type de primitive concerne l'information de texture (photométrie ou colorimétrie) dans la région de la tête. Le concept de flot coloré correspond au couplage d'une recherche de déplacement 3D théorique du modèle sous la contrainte d'appariement de textures. Cela est possible en associant au modèle 3D une composante de texture de manière à assurer la propagation de l'information liée à la pose 3D d'une image à l'autre.

2.3.3 Principe de la méthode de recalage 3D/2D

Les paramètres de pose du modèle pour l'image n étant connus et le champ de déplacement entre l'image n et l'image $n+1$ étant estimé, le principe de la méthode de recalage 3D/2D est décrit ci-dessous (Figure 2.9).

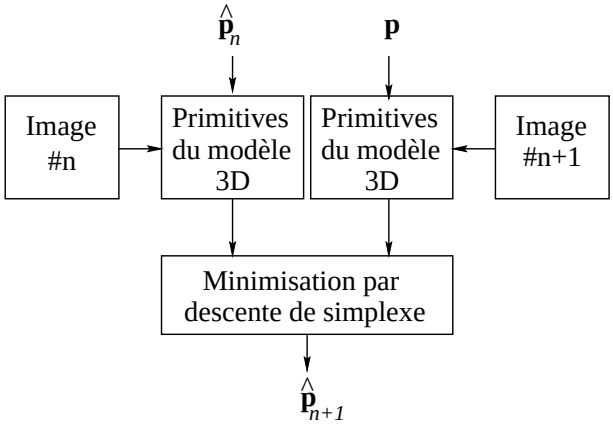


Figure 2.9 : Principe du recalage 3D/2D à base d'un modèle géométrique.

Dans le cas d'une projection parallèle, la pose 3D d'un objet est définie relativement à une position de référence, $\mathbf{p}_{\text{ref}}$, par trois angles de rotation, deux paramètres de translation et un facteur d'échelle. Soit $\mathbf{p}$ le vecteur de pose défini par ces six paramètres, appelé plus simplement *pose*. L'estimation de la pose 3D consiste en la mise à jour itérative du vecteur $\mathbf{p}$ comme suit. Le vecteur de pose associé à l'objet cible dans l'image n, noté $\widehat{\mathbf{p}}_n$, est supposé connu. Une fonction d'erreur ε, préalablement définie, mais dépendant de $\widehat{\mathbf{p}}_n$ et d'une pose arbitraire $\mathbf{p}$, mesure l'écart entre l'ensemble des primitives associées au modèle correspondant d'une part à la pose $\widehat{\mathbf{p}}_n$ et à l'image n, et d'autre part à une pose p et à l'image $n+1$. La nouvelle pose estimée, associée à l'image $n+1$, est définie par la valeur de $\mathbf{p}$ qui minimise ε via la méthode du simplexe, décrite dans la section 2.2.2.

On définit la fonction d'erreur ε de la manière suivante. Soit $\mathcal{M} = \{\mathbf{m}_i,\ i \in \mathcal{J}_{\text{ref}}\}$ un sous-ensemble des points de la surface du modèle dans la position de référence $\mathbf{p}_{\text{ref}}$, $\mathcal{J}_{\text{ref}}$ étant un ensemble d'indexation de ces points. Les points $\mathbf{m}_i$ sont choisis tels que leur projection sur le plan de l'image assure une densité spatiale uniforme et au moins ‹‹pixélique››. Soient $\tau_{\widehat{\mathbf{p}}_n}$ la transformation géométrique qui fait passer le modèle de la position de référence $\mathbf{p}_{\text{ref}}$ à la pose $\widehat{\mathbf{p}}_n$ et π la projection 3D/2D (dans notre cas la projection parallèle). Soit $\mathcal{J}_n \subset \mathcal{J}_{\text{ref}}$ l'ensemble d'indexation des points visibles de la surface du modèle dans la pose $\widehat{\mathbf{p}}_n$. La fonction d'erreur combine une composante relative au mouvement, $\varepsilon_{\text{déplacement}}$ et une autre à la texture, $\varepsilon_{\text{texture}}$.

$\varepsilon_{\text{déplacement}}$ mesure, sur l'ensemble $\mathcal{J}_n$, l'écart entre les champs de déplacement, estimé par l'algorithme de flot optique, noté $\widehat{\mathbf{v}}$, et théorique. Ce dernier est la projection du mouvement de l'objet de la pose $\widehat{\mathbf{p}}_n$ à la pose $\mathbf{p}$, d'où :

$$\varepsilon_{\text{déplacement}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} \left\| \widehat{\mathbf{v}}(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - \pi(\tau_{\mathbf{p}}(\mathbf{m}_i) - \tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i)) \right\| \ , \tag{2.14}$$

où $||\cdot||$ désigne la norme L_1. L'utilisation de la norme L_1 est justifiée par l'observation, expérimentalement constatée, qu'elle conduit aux estimés de pose les plus robustes. Le fait qu'elle n'est pas dérivable à l'origine ne perturbe pas le bon fonctionnement de l'algorithme de minimisation car celui-ci n'utilise pas de dérivées.

$\varepsilon_{\text{texture}}$ exprime l'écart entre la texture de l'image n, projetée sur le modèle 3D, et reprojetée en 2D d'une part et la texture de l'image $n+1$ d'autre part. En notant I_n et I_{n+1} les images n et $n+1$, $\varepsilon_{\text{texture}}$ s'exprime de la manière suivante :

$$\varepsilon_{\text{texture}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} \left| I_n(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - I_{n+1}(\pi(\tau_{\mathbf{p}}(\mathbf{m}_i))) \right| \ . \tag{2.15}$$

La fonction d'erreur ε est une combinaison linéaire de ces deux composantes :

$$\varepsilon = a\,\varepsilon_{\text{texture}} + b\,\varepsilon_{\text{déplacement}}, \text{ où } a,\ b \in \mathbf{R}^+ \tag{2.16}$$

Le rapport a/b sera fixé expérimentalement. L'évaluation de $\varepsilon_{\text{texture}}$ et $\varepsilon_{\text{déplacement}}$ nécessite le calcul des valeurs inter-pixel de l'image et du champ de déplacements, respectivement. Dans la méthode proposée, ces valeurs sont évaluées à l'aide d'une interpolation bilinéaire.

2.3.4 Estimation en présence de grande translation : compensation du mouvement translationnel dominant

Sur le plan théorique, une approche fondée sur l'estimation du champ de déplacements voue la méthode à l'échec pour des translations d'amplitude supérieure à la valeur imposée par les

limitations de l'algorithme de flot (typiquement une vingtaine de pixels pour l'algorithme de Quénot appliqué à des images de taille usuelle).

Pour surmonter cet obstacle, nous avons introduit une méthode de compensation du mouvement translationnel dominant fondée sur une approche de type appariement par blocs. Explicitement, elle revient à calculer le vecteur 2D de translation qui minimise globalement l'expression suivante :

$$\varepsilon_{\text{appariement bloc}}(\mathbf{x}) = \sum_{i \in \mathcal{J}_n} \left| I_{n+1}(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i + \mathbf{x}))) - I_n(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) \right| \ .$$

Cette minimisation est effectuée par une recherche exhaustive dans un voisinage suffisamment étendu de l'origine (en pratique 40×40). Cet appariement par bloc transforme la restriction l'image n à la région du visage en une image compensée en translation, relativement proche de l'image $n+1$. Le champ de déplacement $\widehat{\mathbf{v}}_{\mathbf{c}}$, estimé entre cette image compensée et l'image $n+1$ est plus vraisemblable qu'une estimation directe entre les images n et $n+1$ Le principe de ce traitement est illustré sur Figure 2.10.

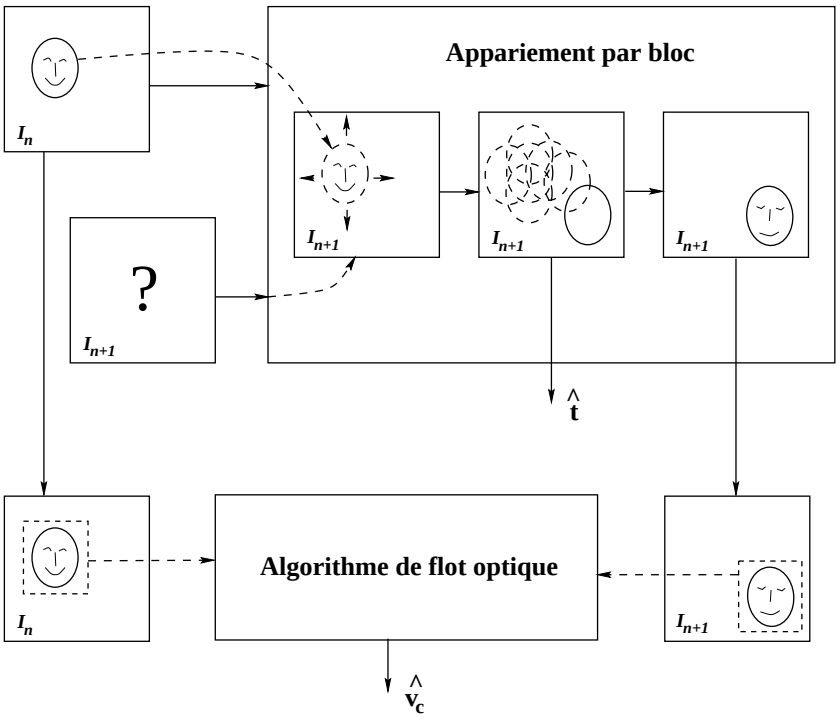


Figure 2.10 : Le principe de compensation du mouvement translationnel dominant.

Si $\widehat{\mathbf{t}}$ est le vecteur de translation dominante ainsi obtenu, les formules de calcul (2.14) et

(2.15) des deux composantes d'erreur, $\varepsilon_{\text{déplacement}}$ et $\varepsilon_{\text{texture}}$ s'expriment comme suit :

$$\varepsilon_{\text{déplacement}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} \left\| \widehat{\mathbf{v}}_{\text{c}}(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - \pi(\tau_{\mathbf{p}}(\mathbf{m}_i) - \tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i)) \right\| \ , \tag{2.17}$$

$$\varepsilon_{\text{texture}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} \left| I_n(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - I_{n+1}(\pi(\tau_{\mathbf{p}}(\mathbf{m}_i)) + \widehat{\mathbf{t}}) \right| \ . \tag{2.18}$$

Dans cette nouvelle formulation, la pose $\widehat{\mathbf{p}}_{n+1}$ s'obtient en additionnant les éléments de $\widehat{\mathbf{t}}$ aux composantes de translation du vecteur $\mathbf{p}$ issu de la minimisation de (2.16).

La Figure 2.11 (c) illustre un exemple d'estimation directe du champ de déplacements $\widehat{\mathbf{v}}$ en présence de grande translation (approximativement 30 pixels), en appliquant l'algorithme de Quénot sur les images montrées Figure 2.11 (a) et (b). Utilisant le même algorithme mais couplé avec la compensation du mouvement translationnel dominant, le champ de déplacements $\widehat{\mathbf{v}}_{\text{c}}$ obtenu (Figure 2.11 (d)) indique sans équivoque le type de mouvement prédominant sur la région du visage, de zoom avant dû au rapprochement de la caméra. De plus, les Figures 2.11 (f) et (g) présentent les résultats du recalage 3D à base de flot optique coloré, à partir de l'initialisation montrée Figure 2.11 (e) et en utilisant le champ de déplacements brut et compensé en translation, respectivement. L' échec est évident dans le premier cas.

2.3.5 Résultats

Pour évaluer objectivement la précision de la pose 3D estimée par l'algorithme ci-dessus, nous avons créé trois séquences correspondant à l'animation d'une tête artificielle, maillée, avec placage de texture de visage réaliste, les paramètres de pose étant définis à partir des estimées de pose 3D des séquences naturelles (corpus de 600 images) comportant différents types de mouvement :

- "Sorin" - rotations et translations de faible vitesse et d'amplitude modérée ;
- "Corneliu" - rotations et translations de vitesse modérée et de grande amplitude ;
- "Armel" - rotations et translations de grandes vitesse et amplitude.

Les images 2D ont été obtenues par projection sur un fond lui-même texturé et mobile. La Figure 2.12 montre quelques images de test ainsi construites.

La Figure 2.13 illustre les fonctions de distributions des erreurs d'estimation pour les six paramètres de la pose 3D correspondant aux trois séquences de test. Les courbes en pointillés correspondent à l'estimation effectuée à base de la composante de texture uniquement ($a = 1$ et $b = 0$ dans l'expression de la fonction d'erreur (2.16)) et celles en trait continu correspondent à l'estimation effectuée sur le flot optique coloré. Le rapport des coefficients de pondération a/b

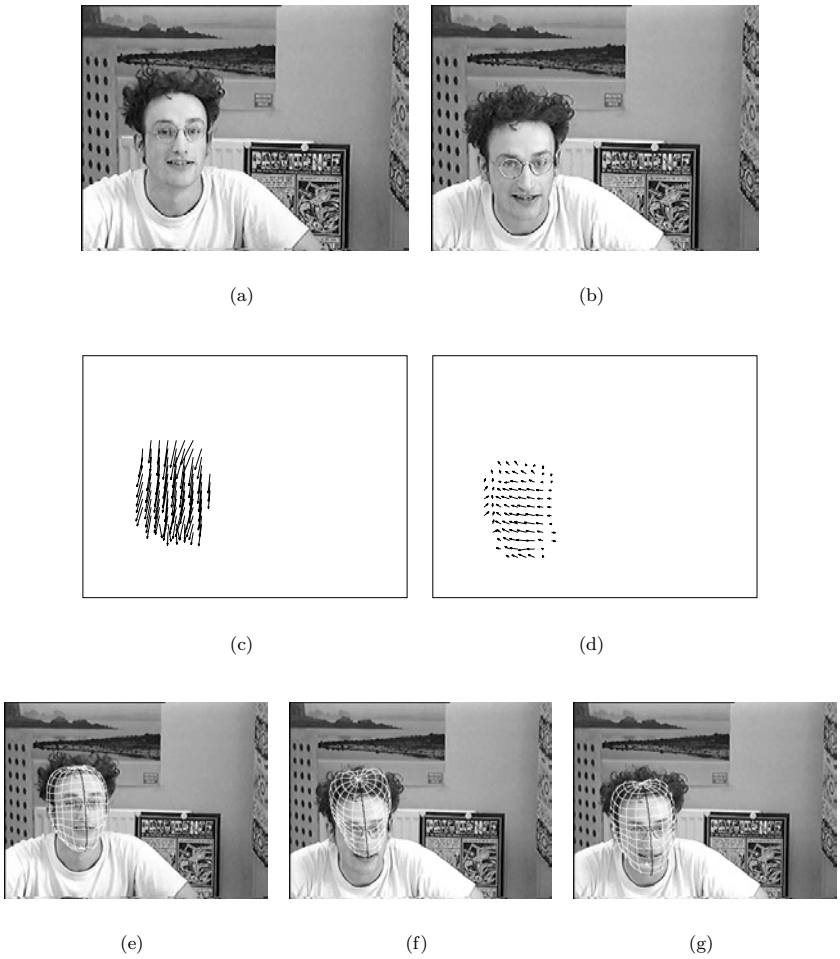
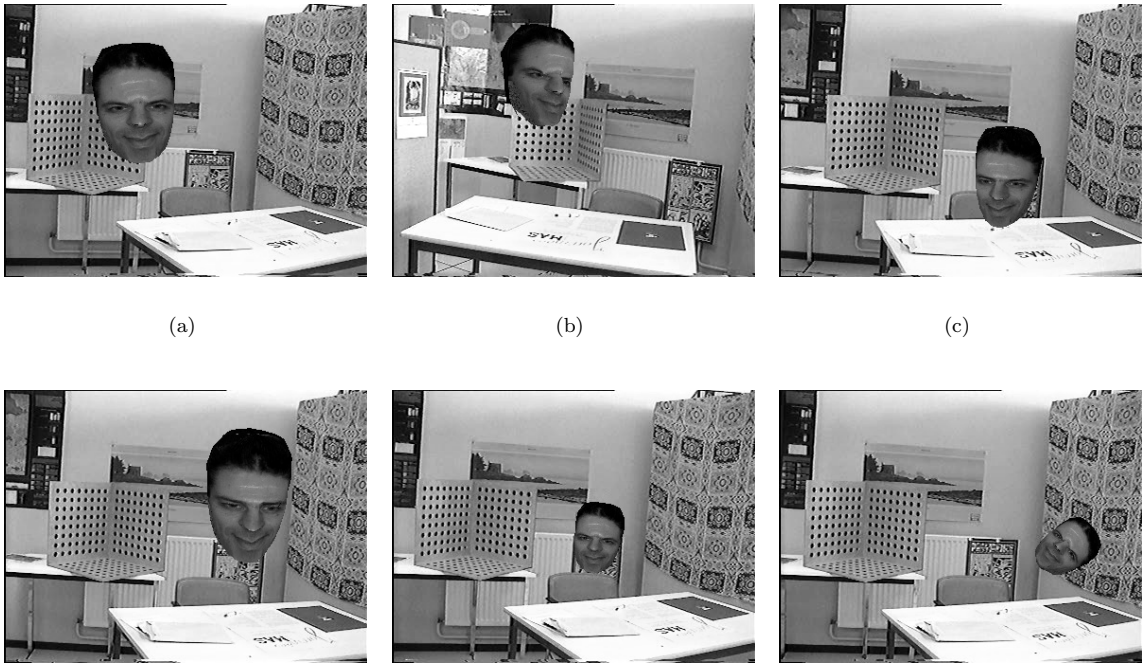


Figure 2.11 : Compensation du mouvement translationnel dominant : (a) et (b) images de test ; (c) champ de déplacements brut ; (d) champ de déplacements estimé après compensation en translation ; (e) initialisation pour recalage 3D effectué sur (a) ; (f) et (g) résultats du recalage 3D à base de flot coloré en utilisant (c) et (d), respectivement. Noter la situation d'échec en (f).

a été fixé à 2 après étude expérimentale de l'influence relative des deux composantes d'erreur (2.17) et (2.18). L'estimation effectuée sur la composante de déplacement uniquement ($a = 0$ et $b = 1$) conduit au décrochage irréversible du modèle après quelques dizaines d'images. Dans la majorité des cas, l'estimation à base de flot coloré fournit des résultats meilleurs que l'estimation utilisant la texture uniquement. ε_α, ε_β et ε_γ sont exprimés en degrés et représentent les erreurs absolues d'estimation correspondant aux angles α, β, γ de rotation par rapport aux axes z, x et y, respectivement, dans un système de coordonnées ayant les axes x et y dans le plan de l'image (i.e. γ et β sont respectivement les angles d'azimut et d'élévation et α est l'angle de rotation dans le plan de l'image). ε_{t_x} et ε_{t_y} sont les erreurs absolues d'estimation des translations t_x et t_y, normalisées aux dimensions horizontale et verticale de la tête et exprimées



(a) (b) (c)

(d) (e) (f)

Figure 2.12 : Images provenant des séquences de test.

en pourcents. ε_s représente l'erreur relative d'estimation du facteur d'échelle s, exprimée en pourcents. Remarquons que les erreurs d'estimation correspondant aux rotations qui gênèrent les auto-occultations de la tête, notamment l'azimut γ et l'élévation β sont sensiblement plus importantes que l'erreur d'estimation de la rotation dans le plan de l'image. Les aspects liés à l'amélioration de l'estimé en présence des auto-occultations seront traités dans la section 3.3.

L'analyse de la distribution des erreurs montre que dans 90% des cas la rotation dans le plan de l'image est estimée à 2° près au plus et que les deux autres rotations le sont à 8° près au plus. En même temps, les translations sont estimées à 3% près au plus de la taille de la tête et le facteur d'échelle est calculé avec une précision relative supérieure à 90% pour toutes les séquences de test.

La méthode d'estimation à base de flot coloré étant ainsi validée, elle a été ensuite appliquée sur des données réelles. La Figure 2.14 illustre le suivi 3D effectué avec succès sur trois séquences vidéos, "Foreman", "Carphone" et "Sorin". Toutes ces séquences comportent des mouvements de vitesse modérée et se caractérisent par l'absence d'occultation. Dans le cas où ces contraintes sont violées, les expérimentations montrent que le risque de décrochage devient important. La Figure 2.15 illustre les limitations de l'estimation de la pose 3D du visage par flot coloré,

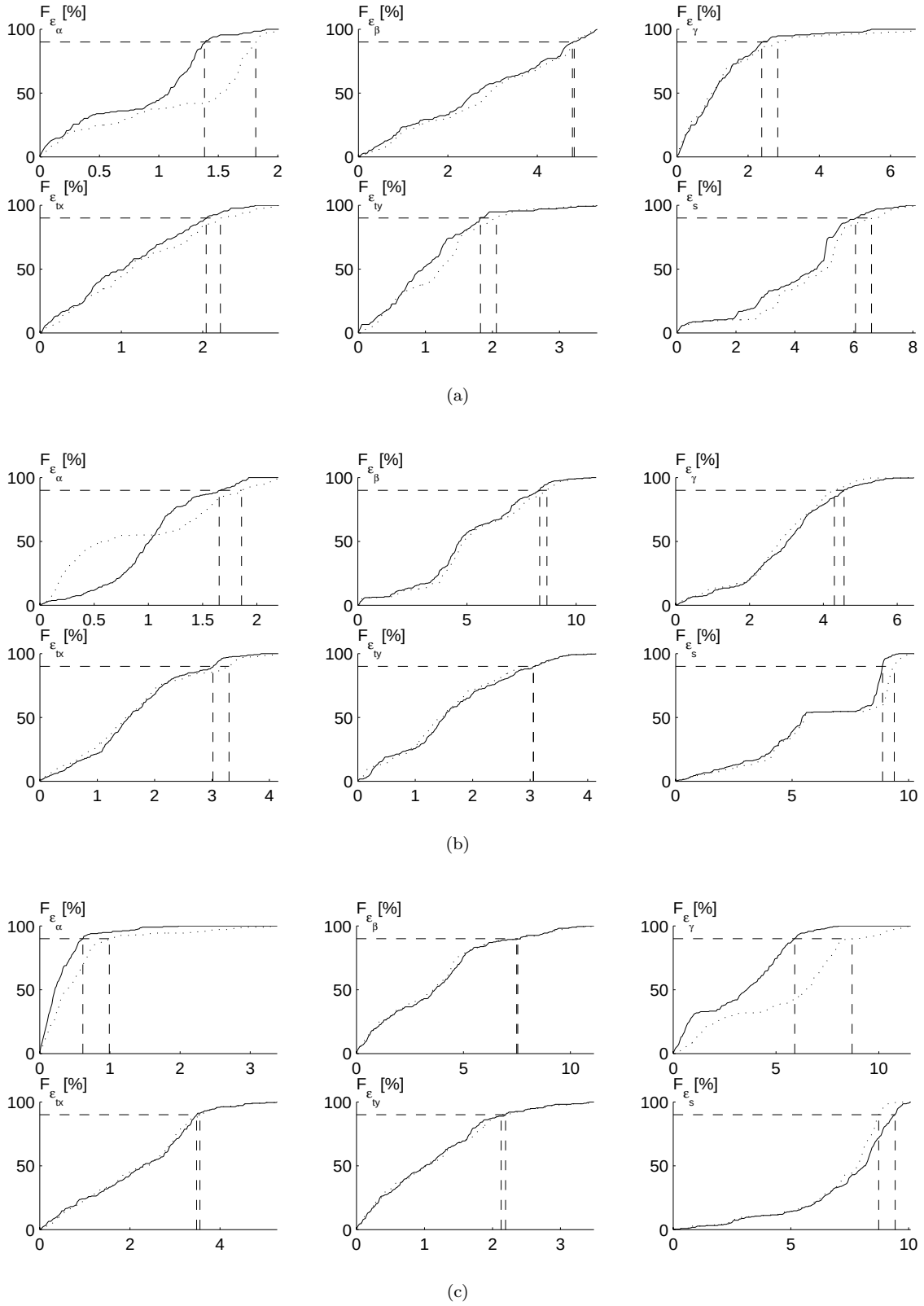
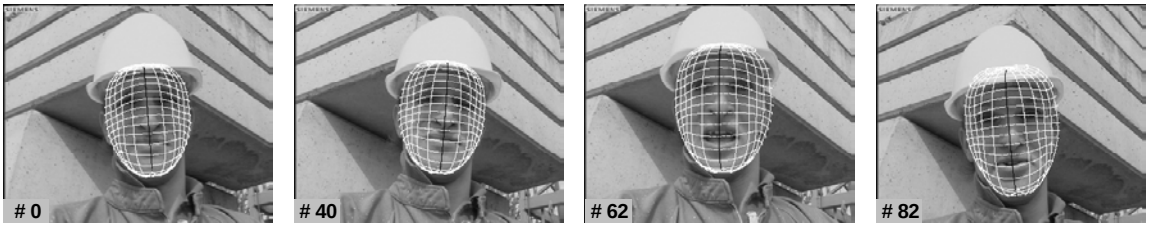


Figure 2.13 : Fonctions de distribution des erreurs d'estimation pour les six paramètres de la pose 3D correspondant aux séquences synthétiques "Sorin" (a), "Corneliu" (b) et Armel (c).

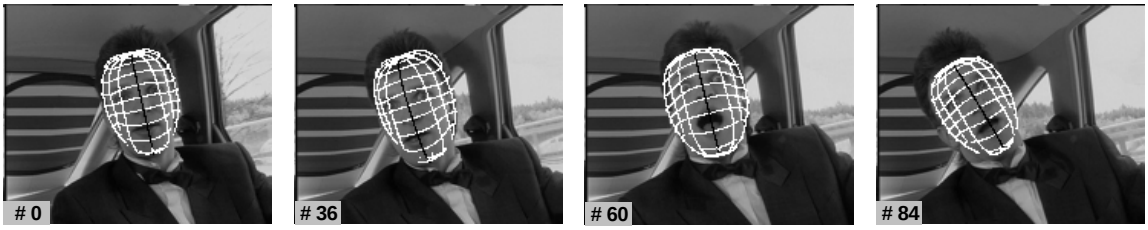
notamment dans les situations suivantes :

- grande vitesse de rotation azimutale ou d'élévation de la tête ;
- présence d'occultations partielles du visage.

Dans le chapitre suivant, nous proposons des apports méthodologiques pour nous affranchir de ces différents problèmes.



(a)



(b)

(c)

Figure 2.14 : La méthode par flot coloré : suivi 3D effectué avec succès sur les séquences "Foreman" (a), "Carphone" (b) et "Sorin" (c).

2.4 Conclusion

Dans ce chapitre nous avons décrit une méthode générique d'estimation de la pose 3D globale de la tête dans des séquences vidéos monoscopiques et non calibrées. L'approche adoptée consiste

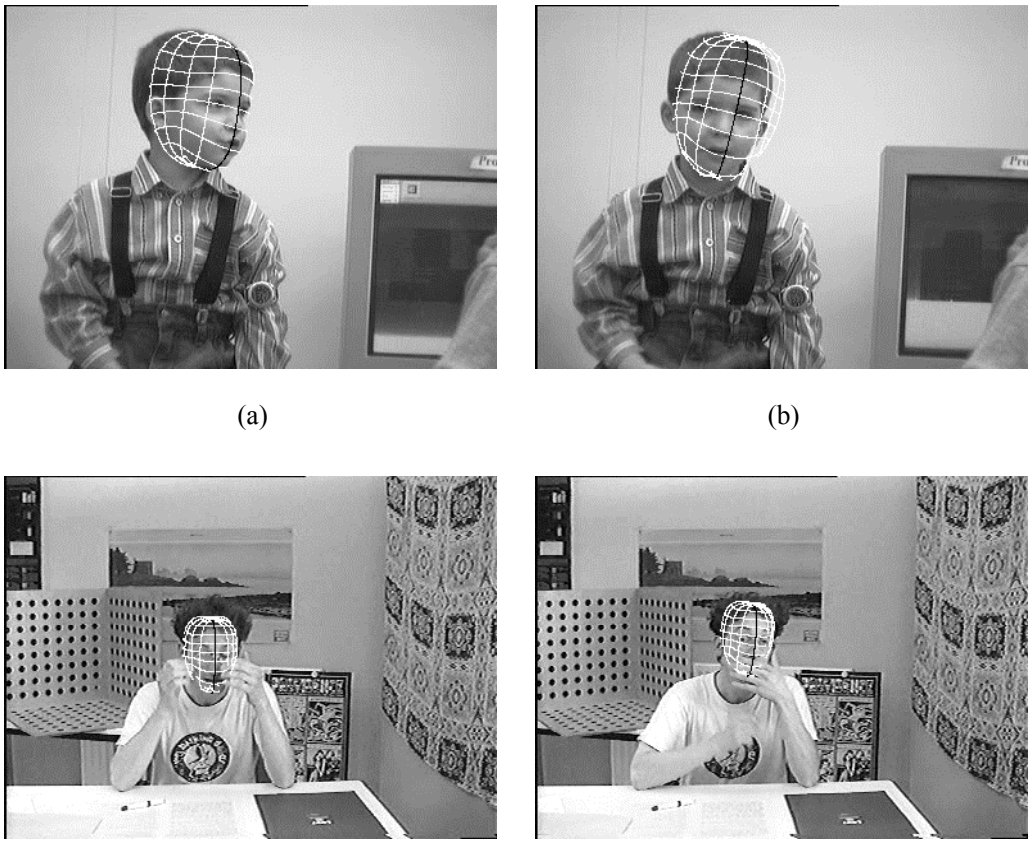


Figure 2.15 : La méthode par flot coloré : échec du suivi 3D dans les situations suivantes : (a) et (b) pour de grandes rotations, (c) et (d) en présence d'occultations partielles du visage.

en une mise en correspondance de primitives 3D d'un modèle de tête avec des primitives 2D extraites des images.

Tout d'abord nous avons présenté une méthode de synthèse d'un modèle générique de tête par une approche analytique et discuté son adéquation à notre problématique par rapport aux techniques de représentation par maillage.

Par la suite, nous avons défini les ensembles de primitives 3D du modèle et 2D des images. Ces primitives sont en correspondance avec des informations de mouvement au travers du champ de déplacements et de texture via la photométrie.

Nous avons évoqué les hypothèses sur lesquelles reposent l'estimation du champ de déplacements et analysé en termes de performance et de précision les approches différentielles intégrées dans un schéma multirésolution et celles de type appariement par blocs en contexte monoréso-

lution. De cette analyse comparée et des résultats expérimentaux obtenus, nous avons justifié le choix d'une approche d'estimation du flot optique par programmation dynamique orthogonale.

Nous avons déduit le concept de flot coloré par introduction des informations de texture dans la région de la tête et nous avons mis en œuvre les primitives définies pour réaliser le recalage 3D/2D du modèle de tête sur les images. Le principe de la mise en correspondance repose sur la minimisation par la méthode du simplexe d'une fonctionnelle d'erreur composite, définie de manière à assurer la propagation d'une image à l'autre, de l'information liée à la pose 3D du modèle, à travers le flot coloré.

Afin de pallier les limitations bien connues d'estimation du flot optique, nous avons introduit la compensation en mouvement translationnel dominant, mettant en œuvre une approche de type appariement par blocs.

Enfin, les résultats obtenus sur des séquences synthétiques et réelles ont été discutés et les cas d'échec analysés. Sur des séquences calibrées de synthèse, l'estimation est effectuée dans 90% des cas avec une précision de 2° pour la rotation de la tête dans le plan de l'image, de 8° pour les rotations qui génèrent des auto-occultations, de 97% (relativement à la taille de la tête) pour les translations et de 90% pour le facteur d'échelle.

Chapitre 3

Estimation robuste de la pose 3D du visage

3.1 Avant-propos

Dans le chapitre précédent, nous avons rencontré un certain nombre de limitations que l'estimation par flot coloré ne peut surmonter. Le problème lié aux grandes translations a toutefois été résolu par une approche de compensation du mouvement translationnel dominant, conduisant à postuler que le flot optique estimé l'est de manière robuste.

Dans ce contexte, des apports méthodologiques sont proposés ici pour s'affranchir des problèmes posés par les mouvements complexes de grande vitesse angulaire, ainsi que par les occultations occasionnelles.

Disposant du champ de déplacements compensé en translation dominante, nous développons une technique d'interpolation spatio-temporelle fondée sur une modélisation ondulatoire hiérarchique dans le cadre d'une approche physique par groupe de paquets d'onde. A l'aide de cette interpolation, nous sommes en mesure de synthétiser des images virtuelles afin de guider un recalage 3D/2D stable et précis en présence de grandes rotations. En outre, nous introduisons un critère de visibilité issu des principes d'estimation robuste permettant d'améliorer l'estimé de la pose 3D en présence d'auto-occultations de la tête.

Ensuite, nous montrons comment les cas d'occultation du visage par un objet présentant un mouvement relatif par rapport à celui-ci sont gérés en considérant une classification fondée sur un critère de mouvement sous contrainte de régularité spatiale, couplée avec une analyse de similarité de mouvements à base de modèle paramétrique.

Enfin, les simulations numériques sur les séquences calibrées de synthèse sont reprises, pour illustrer la pertinence des approches développées et de nouveaux résultats obtenus sur des séquences réelles sont présentés et discutés.

3.2 Interpolation temporelle : approche physique par groupe de paquets d'onde

Parmi les trois familles de méthodes permettant d'interpoler temporellement deux images (technique bien connue sous le nom de *morphing*) la première [Beier92] est fondée sur l'utilisation de primitives géométriques, la seconde [Hughes92] privilégie la représentation fréquentielle et la troisième [Quénot97] s'appuie sur la connaissance du champ de déplacement. Cette dernière est donc bien adaptée à notre méthodologie puisque nous disposons déjà de l'estimation du flot optique.

3.2.1 Interpolation linéaire

Soient $I_0(\mathbf{x})$ et $I_1(\mathbf{x})$ deux images similaires en terme de scène (contenu et fond) et $\alpha \in [0,\, 1]$ un nombre réel. Ici, $\mathbf{x} = (x,\, y)$ représente les coordonnées spatiales bidimensionnelles. Supposons qu'on veuille synthétiser une image I_α correspondant à I_0 ou à I_1 si α vaut respectivement 0 ou 1.

La technique la plus simple consiste en une interpolation linéaire du champ de déplacements estimé entre les images I_0 et I_1. A tout pixel $\mathbf{x}$ de I_α, on associe son vecteur déplacement $\mathbf{v}_{01}(\mathbf{x})$, de sorte que le barycentre des extrémités de son support affectées des coefficients α et $(1-\alpha)$ coïncide avec $\mathbf{x}$.

Les extrémités de ce vecteur translaté de $-\alpha\, \|\mathbf{v}_{01}(\mathbf{x})\|$ ne correspondent en général pas à des points de discrétisation, d'où la nécessité, pour définir leur intensité, de procéder à une interpolation bilinéaire en fonction des intensité dans I_0 pour l'origine du vecteur et dans I_1 pour son extrémité (Figure 3.1).

Formellement, avec les notations de la Figure 3.1, en posant :

$$\begin{aligned}
A &= (1-u_1)\,(1-v_1)\, I_1(x_1,\, y_1) + u_1\,(1-v_1)\, I_1(x_1+1,\, y_1) + \\
&\quad (1-u_1)\, v_1\, I_1(x_1,\, y_1+1) + u_1\, v_1\, I_1(x_1+1,\, y_1+1) \\
B &= (1-u_0)\,(1-v_0)\, I_0(x_0,\, y_0) + u_0\,(1-v_0)\, I_0(x_0+1,\, y_0) + \\
&\quad (1-u_0)\, v_0\, I_0(x_0,\, y_0+1) + u_0\, v_0\, I_0(x_0+1,\, y_0+1)\ ,
\end{aligned}$$

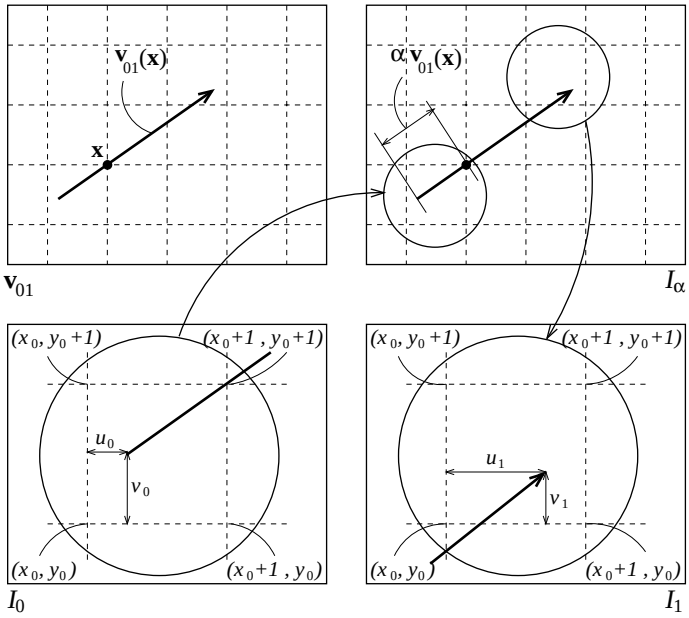


Figure 3.1 : Schéma de l'interpolation simple du flot.

les valeurs interpolées bilinéairement donnent l'intensité des extrémités du vecteur déplacement translaté. L'intensité du pixel $\mathbf{x}$ à la date α s'exprime par :

$$I_\alpha(x, y) = \alpha A + (1 - \alpha) B \ .$$

Cette méthode aboutit à des résultats acceptables pour des images assez semblables (Figure 3.2). Toutefois, l'intensité en un pixel n'est imposée que par les valeurs de l'intensité aux deux extrémités du vecteur déplacement et ne prend pas en compte l'information au voisinage de ce pixel. Cette faiblesse conduit à une mauvaise interpolation dans le cas où les deux images sont dissemblables, particulièrement au niveau des contours d'objets, pour lesquels les transitions ne sont pas respectées (Figure 3.2).

3.2.2 Analogie ondulatoire : le modèle du front d'onde

Les cas d'échec précédemment soulignés, nous conduisent à introduire une discontinuité dans l'interpolation pour favoriser un basculement d'une information type "fond" à une information type "objet".

Pour cela nous proposons un modèle de propagation par front d'onde. Le principe consiste à affecter à un pixel donné l'intensité de la source à l'origine de l'onde qui l'atteint en premier. Tous les pixels se comportent comme des sources ponctuelles émettant des ondes de célérité

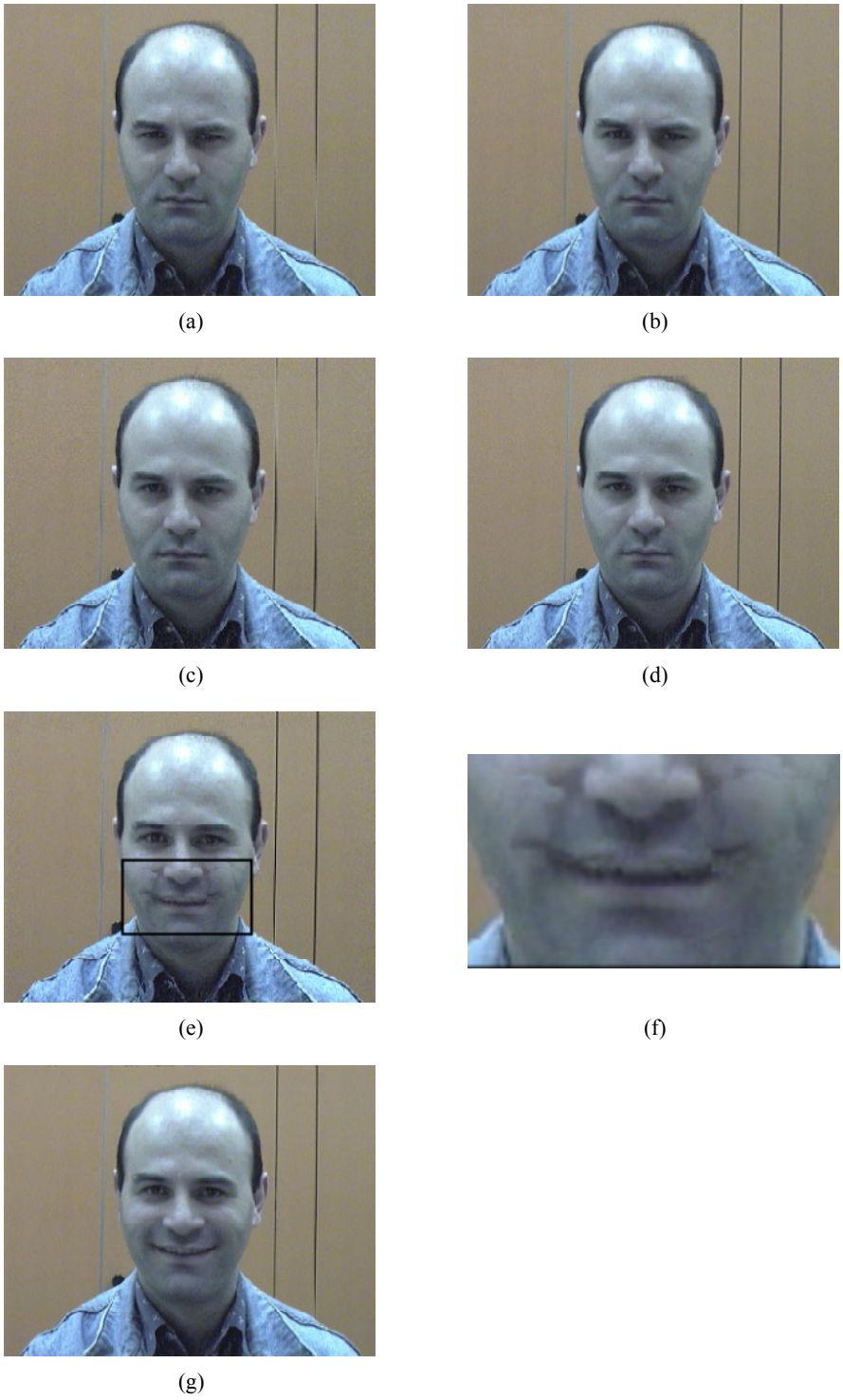


Figure 3.2 : Interpolation linéaire. Dans le cadre de petits déplacements entre les images originales (a) et (d), les résultats (b) et (c) sont satisfaisants. Pour d'importants déplacements entre les images originales (d) et (g), l'interpolée (e) et le zoom (f) de la région encadrée montrent les limites de la méthode.

différentes définies en tout point par :

$$c(\mathbf{x}) = \frac{\|\mathbf{v}_{01}(\mathbf{x})\|}{\tau} ,$$

où $\mathbf{v}_{01}(\mathbf{x})$ est le champ de déplacement au point $\mathbf{x}$, et τ le temps qui sépare les deux images I_0 et I_1.

Le vecteur d'onde est porté par le vecteur déplacement au point source. Cette onde propage une information d'intensité du pixel source, en raison de l'hypothèse de stationnarité des intensités. Les ondes peuvent être progressives, lorsque la source appartient à l'image I_0, ou régressives lorsque la source appartient à l'image I_1.

La principale faiblesse de ce type de propagation réside dans son instabilité, due aux erreurs d'estimation du champ de déplacement, et dans son manque de fiabilité, par le choix, pour chaque pixel, d'une unique source d'information.

Rendre robuste la modélisation précédente nécessite de régulariser les discontinuités introduites par le front d'onde, alors que la rendre fiable requiert de multiplier les sources susceptibles d'influencer l'intensité du pixel.

Pour satisfaire à la première propriété, nous proposons une modélisation par paquets d'onde. Introduisons tout d'abord intuitivement cette notion à partir d'un exemple. Dans une course de relais, où le front d'onde est symbolisé par le témoin, le coureur qui attend de le recevoir commence à courir avant même d'être en possession de celui-ci. De même, le coureur qui transmet le témoin continue de courir après le passage du relais. Cette anticipation et cette hystérésis expriment l'information avant et après le passage du front d'onde et constitue ce que les physiciens appellent un paquet d'onde. Autrement dit, un point n'a pas besoin d'avoir été atteint par le front d'onde pour recevoir une partie de l'information.

Par ailleurs, l'hypothèse de stationnarité de l'intensité nous conduit à formuler cette analogie dans un contexte non altéré, *i.e.* pour une information transmise et non modifiée sur son parcours. Cela revient à supposer que le milieu dans lequel se propagent les ondes n'est pas absorbant : les paquets d'onde se propagent sans se déformer.

La condition de fiabilité est remplie dès lors qu'on considère plusieurs sources pouvant émettre chacune un paquet d'onde. Cela revient à modéliser les effets d'un groupe de paquets d'onde.

3.2.3 Modélisation par groupe de paquets d'onde

On modélise la qualité d'information reçue comme étant inversement proportionnelle à la distance au front d'onde, le coefficient de proportionnalité étant un facteur de normalisation

égal à la somme des inverses des distances à tous les fronts d'onde (Figure 3.3).

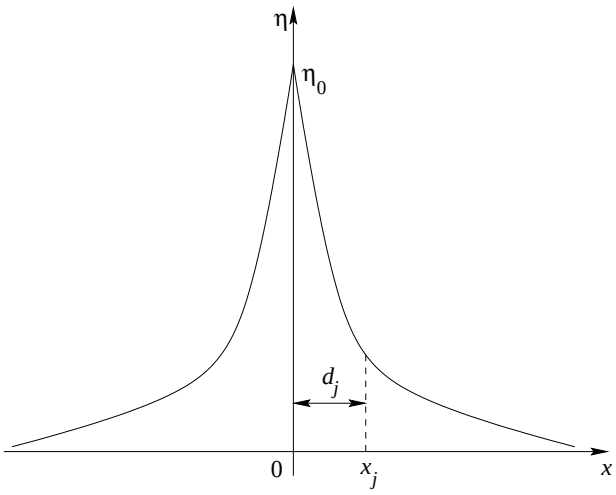


Figure 3.3 : Front d'onde localisé à l'origine. Son amplitude est égale à l'intensité du point source dont est issue l'onde. A distance du front, la quantité d'information reçue est réduite d'un facteur proportionnel à l'inverse de cette distance.

L'expression de l'intensité en $\mathbf{x}$ peut s'écrire grâce au théorème de superposition (puisque les équations de propagation sont linéaires) comme la somme des différentes informations reçues en $\mathbf{x}$ de la part de tous les paquets d'onde dont les fronts d'onde.

Notons d_i l'ensemble des distances de $\mathbf{x}$ aux différents fronts d'onde et η_j l'information transmise par le front d'onde j. L'information reçue en x de la part du front d'onde i est :

$$\eta_i'(\mathbf{x}) = \frac{\eta_i \dfrac{1}{d_i}}{\displaystyle\sum_j \frac{1}{d_j}} \ .$$

L'information totale reçue de la part de tous les fronts d'onde s'exprime donc par :

$$\eta(\mathbf{x}) = \sum_i \eta_i'(\mathbf{x}) \ .$$

Cependant, puisque l'atténuation de l'information est une fonction de l'inverse de la distance au front d'onde, on peut approcher cette expression en ne sommant que sur les fronts d'onde les plus proches de $\mathbf{x}$, ce que nous formalisons ci-dessous.

Calculons les champs de déplacement $\mathbf{v}_{01}(\mathbf{x})$ et $\mathbf{v}_{10}(\mathbf{x})$ respectivement entre I_0 et I_1, et I_1 et I_0. L'image intermédiaire I_α est générée pixel par pixel de la manière suivante. Soit Ω l'ensemble des points $\mathbf{x} + \alpha\,\mathbf{v}_{01}(\mathbf{x})$ et $\mathbf{x} + (1-\alpha)\,\mathbf{v}_{10}(\mathbf{x})$. Pour chaque pixel $\mathbf{x}_0$, on définit :

$$r_0 = \inf\left\{ r \mid \mathrm{Card}\left(\mathcal{B}_r(\mathbf{x}_0) \cap \Omega\right) \geq k \right\} \ ,$$

où $\mathcal{B}_r(\mathbf{x}_0)$ est la boule de centre $\mathbf{x}_0$ et de rayon r et k un entier non nul donné (en pratique, $k = 3$ représente un bon compromis entre la qualité du morphing et le temps de calcul).

Les éléments $\mathcal{B}_{r_0}(\mathbf{x}_0) \cap \Omega$, de cardinal k_0, sont notés $y_1, y_2, \ldots, y_{k_0}$ et leurs antécédents $x_1, x_2, \ldots, x_{k_0}$. Soit $l(i)$ la fonction indicatrice de l'origine de l'antécédent, définie par :

$$\forall\, i \in \{1, 2, \ldots, k_0\}\,,\; l(i) = \begin{cases} 0, \text{ si } \mathbf{y}_i = \mathbf{x}_i + \alpha\, \mathbf{v}_{01}(\mathbf{x}) \\ 1, \text{ si } \mathbf{y}_i = \mathbf{x}_i + (1-\alpha)\, \mathbf{v}_{10}(\mathbf{x}) \end{cases} .$$

L'image intermédiaire I_α au point $\mathbf{x}_0$ est exprimée par la combinaison linéaire suivante :

$$I_\alpha(\mathbf{x}_0) = \frac{\displaystyle\sum_{i=1}^{k_0} \frac{1}{d(\mathbf{x}_0, \mathbf{y}_i)}\, I_{l(i)}(\mathbf{x}_i)}{\displaystyle\sum_{i=1}^{k_0} \frac{1}{d(\mathbf{x}_0, \mathbf{y}_i)}} ,$$

$d(\cdot,\cdot)$ étant la distance d_1 (4-connexité) ou d_2 (distance euclidienne). Les expérimentations montrent que la qualité du morphing est pratiquement indépendante de la distance utilisée.

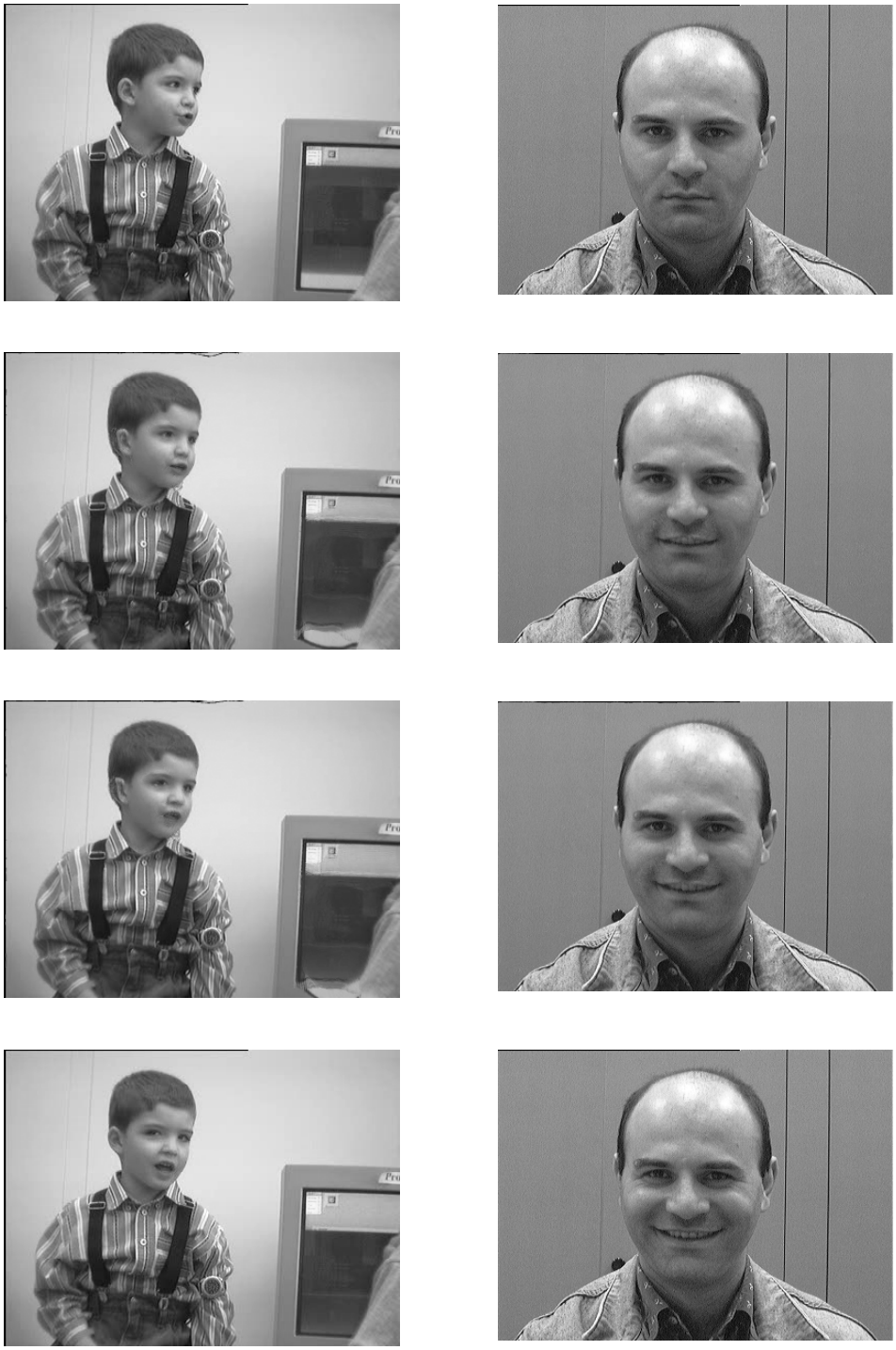
Les images interpolées par cette modélisation, dans le cadre de mouvements rigides et de déformations, sont présentées Figure 3.4. Les images virtuelles créées à partir de cette modélisation sont suffisamment précises et réalistes pour servir d'intermédiaires lors de la phase de recalage 3D/2D afin de prendre en compte des rotations de grande magnitude ou des déformations locales importantes. Afin de garantir la robustesse de l'interpolation temporelle, celle-ci est effectuée à partir du flot optique compensé en composante translationnelle dominante.

Toutefois, pour minimiser le temps de calcul, le morphing ne sera déclenché que dans le cas où le point de minimum détecté au cours de la procédure d'optimisation de la fonction d'erreur ε correspond à une valeur trop importante de celle-ci. En effet, les valeurs de minimum de ε peuvent être appréhendées au cours du recalage le long de la séquence.

3.3 Estimation robuste à base d'indice de visibilité

Dans la section 2.3.5 nous avons remarqué une disproportion assez importante entre la précision du suivi de la rotation dans le plan de l'image, d'une part, et les rotations azimutales et d'élévation d'autre part. Cet effet est dû principalement au fait que ces deux dernières rotations génèrent des auto-occultations de la tête.

En nous inspirant des principes d'estimation robuste [Hubert81], notamment en ce qui concerne la limitation de l'influence des *outliers*, nous définissons un nouveau critère d'estimation qui prend en compte la géométrie 3D du modèle. Afin de réduire l'influence des points



(a) (a)

Figure 3.4 : Interpolation par groupe de paquets d'onde entre les images originales (1ère et dernière ligne) pour des rotations (plus de 15°) - (a) et des expressions faciales (b). Les résultats (b) sont de bien meilleure qualité que sur la Figure 3.2 (e).

susceptibles de disparaître suite aux effets de rotation azimutales et d'élévation entre deux images, nous pondérons, dans la fonction d'erreur, tous les points visibles du modèle dans la pose $\widehat{\mathbf{p}}_n$ par un indice de visibilité, noté v et défini par la projection de la normale à la surface du modèle 3D sur l'axe perpendiculaire au plan de l'image. La Figure 3.5 illustre la distribution de cet indice de visibilité sur la projection du modèle, relativement à la position de référence. Les pixels situés au voisinage des bords de la projection, où les valeurs de l'indice de visibilité sont relativement faibles, auront ainsi une contribution modeste au recalage.

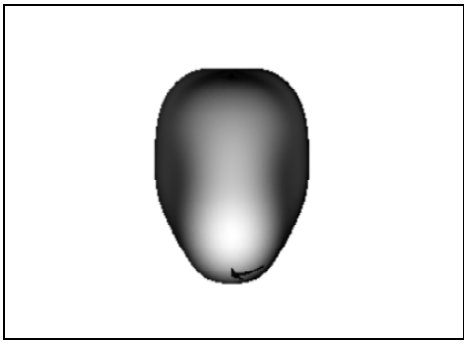


Figure 3.5 : Représentation de l'indice de visibilité du modèle dans la position de référence.

Dans cette nouvelle formulation, les deux composantes d'erreur, $\varepsilon_{\text{déplacement}}$ et $\varepsilon_{\text{texture}}$ données respectivement par (2.17) et (2.18) deviennent :

$$\varepsilon_{\text{déplacement}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} v(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i)) \left\| \widehat{\mathbf{v}}_{\text{c}}(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - \pi(\tau_{\mathbf{p}}(\mathbf{m}_i) - \tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i)) \right\| \ , \quad (3.1)$$

$$\varepsilon_{\text{texture}} = \frac{1}{\text{Card}(\mathcal{J}_n)} \sum_{i \in \mathcal{J}_n} v(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i)) \left| I_n(\pi(\tau_{\widehat{\mathbf{p}}_n}(\mathbf{m}_i))) - I_{n+1}(\pi(\tau_{\mathbf{p}}(\mathbf{m}_i)) + \widehat{\mathbf{t}}) \right| \ . \quad (3.2)$$

Les améliorations apportées par cette technique d'estimation seront discutées dans la section 3.5.

3.4 Analyse des occultations

3.4.1 Positionnement du problème

Dans la section 2.3.5, nous avons montré que les occultations partielles du visage limitent fortement les performances du recalage 3D/2D par flot coloré. Afin de pallier cette limitation, nous proposons ici une technique orientée segmentation au sens du mouvement, pour détecter les occultations du visage par des objets ayant un mouvement relatif par rapport à celui-ci.

La segmentation au sens du mouvement est une technique d'analyse spatio-temporelle des séquences d'images dont le but est d'agréger des ensembles de pixels en exploitant l'homogénéité

du mouvement 2D apparent. Elle diffère de la segmentation classique d'images car elle prend en compte l'information temporelle en plus de l'information spatiale.

Dans le cas d'analyse des mouvements du visage, la segmentation au sens du mouvement permet d'une part de mettre en évidence les éventuelles zones occultées et d'autre part de discriminer les parties rigides des parties déformables. Un estimateur robuste de la pose 3D doit prendre en compte uniquement les régions non occultés et non déformées. Précisons que nous nous référons aux occultations au sens du mouvement, *i.e.* détectables par segmentation au sens du mouvement. Par exemple, une paire de lunettes solidaire du visage ne constitue pas un objet occultant au sens du mouvement. Dans cet exemple, l'objet attaché au visage ne perturbe pas l'estimation de la pose 3D et ainsi il n'est pas nécessaire d'être considéré occultant.

La méthode de segmentation au sens du mouvement présentée ici s'appuie sur deux étapes successives, une classification au sens du mouvement sous contrainte de régularité spatiale et une analyse de similarité de mouvements à base de modèle paramétrique.

La technique de classification au sens du mouvement fait appel à l'algorithme des *k-moyennes* appliqué sur le champ de déplacements calculé dans la région du visage et imposant *a priori* certaines contraintes de régularité spatiale, de manière à renforcer la robustesse. Cette approche a été préférée à d'autres techniques de classification puisqu'elle est peu coûteuse en termes de volume de calcul et qu'elle fournit de bons résultats dans la majorité des situations étudiées. Présentons tout d'abord l'algorithme des *k-moyennes*.

3.4.2 L'algorithme des *k-moyennes*

L'algorithme des *k-moyennes* est une technique générique de classification multidimensionnelle non supervisée en un nombre préétabli de classes. Pour présenter cet algorithme, notons par $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbf{R}^n$ l'ensemble de vecteurs à classifier et par $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K \subset \mathcal{X}$ les classes, disjointes deux à deux, auxquelles les vecteurs $\mathbf{x}_i$ doivent être affectés. Formellement, l'objectif de la classification consiste à établir pour chaque paire vecteur-classe un coefficient d'appartenance $a_{ik} = \begin{cases} 1, & \text{si } \mathbf{x}_i \in \mathcal{C}_k \\ 0, & \text{sinon} \end{cases}$. Dans l'algorithme des *k-moyennes* chaque classe est caractérisée uniquement par la valeur moyenne de ses éléments :

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} a_{ik}\, \mathbf{x}_i}{\sum_{i=1}^{N} a_{ik}} \,. \tag{3.3}$$

Chaque vecteur $\mathbf{x}_i$ est affecté à une classe par rapport à un critère simple de distance :

$$\alpha = \underset{1 \leq k \leq K}{\arg\min}\ \mathrm{dist}(\mathbf{x}_i, \boldsymbol{\mu}_k)\ ,\ a_{ik} = \begin{cases} 1, & \text{si } k = \alpha \\ 0, & \text{sinon} \end{cases}\ ,\ \text{pour } k = 1, 2, \ldots, K\ , \tag{3.4}$$

où $\mathrm{dist}(\cdot, \cdot)$ est une distance usuelle sur $\mathbf{R}^n$ (d_1 ou d_2).

La procédure des *k-moyennes* consiste à mettre à jour alternativement les valeurs des a_{ik} et des $\boldsymbol{\mu}_k$ jusqu'à ce que la stabilité soit obtenue. Explicitement, l'algorithme se déroule comme suit :

1. initialiser $\boldsymbol{\mu}_k$ et après a_{ik} (formule (3.4)) pour $k = 1, 2, \ldots, K$ et $i = 1, 2, \ldots, N$;
2. pour tout $k = 1, 2, \ldots, K$ calculer $\boldsymbol{\mu}_k$ avec la formule (3.3) ;
3. pour tout $i = 1, 2, \ldots, N$ et $k = 1, 2, \ldots, K$ calculer a_{ik}^{new} avec la formule (3.4) ;
4. *si* $\exists i\ \exists k$ tels que $a_{ik}^{\mathrm{new}} \neq a_{ik}$

   $a_{ik} = a_{ik}^{\mathrm{new}}$, pour tous $i = 1, 2, \ldots, N$ et $k = 1, 2, \ldots, K$;
   aller à 2 ;

   sinon

   retourner les a_{ik}^{new} ;

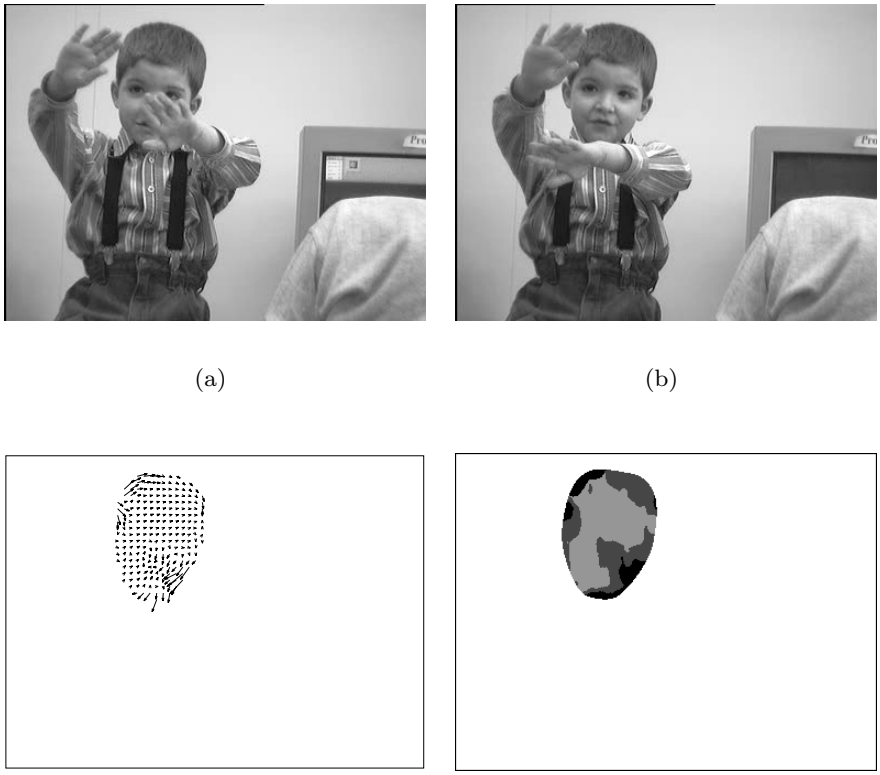
   ARRET ;

Une variante de l'algorithme consiste à initialiser aléatoirement les appartenances a_{ik} et à calculer après les moyennes $\boldsymbol{\mu}_k$. Dans les deux cas, la convergence est obtenue assez rapidement, après quelques dizaines de cycles, par exemple, pour une classification dans $\mathbf{R}^6$.

3.4.3 Classification au sens du mouvement

L' algorithme des *k-moyennes* appliqué directement sur les vecteurs de déplacement ne conduit pas à des résultats robustes de classification au sens du mouvement que dans des cas particuliers, car le flot optique est, en général corrompu par des erreurs dues aux occultations. De plus, l'information liée à la distribution spatiale du champ de déplacement ne doit pas être ignorée. En effet, le résultat de la classification risque d'être alors fragmentaire et ainsi il sera difficile de mettre en évidence les différents objets ou parties d'objets en mouvement (Figure 3.6).

Pour surmonter ces inconvénients, nous avons mis au point une technique de classification capable d'agréger localement l'information contenue dans le flot optique. Dans une première étape de traitement, le champ de déplacement $\mathbf{v}$ de la région à segmenter est localement ajusté



(a) (b)

(c) (d)

Figure 3.6 : L'algorithme des *k-moyennes* appliqué directement aux vecteurs de déplacement (c) calculés entre les images (a) et (b) conduit à un résultat (d) difficile à exploiter.

au sens des moindres carrés par un modèle affine. Cet ajustement est effectué pour chaque pixel $\mathbf{x} = (x,\, y)$, par rapport à un voisinage carré $\mathcal{V}_{\mathbf{x}}$ de taille préétablie (en pratique 15 pixels) :

$$\sum_{\mathbf{y}\in\mathcal{V}_{\mathbf{x}}} \left\| \mathbf{A}\cdot(\mathbf{y}-\mathbf{x})^t + \mathbf{b} - \mathbf{v}^t(\mathbf{y}) \right\|^2 \overset{\text{not}}{=} \varepsilon_{\mathbf{x}}(\mathbf{A},\, \mathbf{b}) : \text{ minimale },$$

où $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ et $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ sont les paramètres du modèle affine, à calculer, et $(\cdot)^t$ représente l'opérateur de transposition. Minimiser $\varepsilon_{\mathbf{x}}$ revient à résoudre le système linéaire :

$$\begin{aligned} \mathbf{0} &= \nabla_{\mathbf{A}}\varepsilon_{\mathbf{x}}(\mathbf{A},\, \mathbf{b}) = \sum_{\mathbf{y}\in\mathcal{V}_{\mathbf{x}}} \left[\mathbf{A}\cdot(\mathbf{y}-\mathbf{x})^t \cdot (\mathbf{y}-\mathbf{x}) + \mathbf{b}\cdot(\mathbf{y}-\mathbf{x}) - \mathbf{v}^t(\mathbf{y})\cdot(\mathbf{y}-\mathbf{x})\right] \\ \mathbf{0} &= \nabla_{\mathbf{b}}\varepsilon_{\mathbf{x}}(\mathbf{A},\, \mathbf{b}) = \sum_{\mathbf{y}\in\mathcal{V}_{\mathbf{x}}} \left[\mathbf{A}\cdot(\mathbf{y}-\mathbf{x})^t + \mathbf{b} - \mathbf{v}^t(\mathbf{y})\right] \end{aligned} , \qquad (3.5)$$

où $\nabla_{\mathbf{A}}$ et $\nabla_{\mathbf{b}}$ sont les gradients par rapport aux éléments de $\mathbf{A}$ et de $\mathbf{b}$, respectivement. Remarquons que les éléments variables par rapport au pixel se retrouvent dans les termes libres du système (3.5). Ainsi, l'estimation des paramètres du modèle affine en un pixel revient à un filt-

rage linéaire (multiplication de l'inverse de la matrice du système (3.5) et un vecteur dépendant de pixel), donc elle est peu coûteuse en volume de calcul.

Les Figures 3.7 (d) et 3.8 (d) montrent la régularisation localement affine obtenue en appliquant cette technique aux champs de déplacements illustrés respectivement Figures 3.7 (c) et 3.8 (c).

La segmentation du visage en régions comportant des mouvements de natures *a priori* différentes consiste à appliquer l'algorithme des *k-moyennes* sur les vecteurs 6-dimensionnels des modèles affines, $\mathbf{m}_i = (a_{11}^{(i)}, a_{12}^{(i)}, a_{21}^{(i)}, a_{22}^{(i)}, b_1^{(i)}, b_2^{(i)})$, associés aux pixels $\mathbf{x}_i$ de la projection du modèle 3D recalé. Les paramètres $\mathbf{m}_i$ sont calculés sur le flot optique compensé en mouvement translationnel dominant.

Les Figures 3.7 (e) et 3.8 (e) présentent deux exemples de classification au sens du mouvement sur la région du visage. Il s'agit d'une occultation partielle de la tête - Figure 3.7 (a) et (b) et d'un visage parlant (déformations dans la région de la bouche) - Figure 3.8 (a) et (b). Dans les deux cas, le nombre de classes est fixé à 3. Comme particularité dans le deuxième exemple, remarquons les régions en gris foncé segmentées sur le bords du visage. Ce phénomène est dû à la corruption du flot optique dans les régions de transition et partiellement au fait que le modèle affine du mouvement sur lequel la classification est fondée représente une bonne approximation pour le mouvement 3D d'un plan projeté en 2D.

3.4.4 Analyse de similarité de mouvements

La classification étant effectuée, on dispose d'un certain nombre de régions, chacune décrite par un modèle affine de mouvement, notamment donné par la moyenne de la classe correspondante. Il faut à présent savoir si les mouvements caractérisant ces régions sont de nature vraiment différente. Pour cela nous avons développé une technique d'analyse de similarité à base de modèle paramétrique de mouvement, en nous inspirant de celle présentée dans [Zaharia01]. Les auteurs développent une méthode permettant de comparer des modèles paramétriques de mouvement par l'intermédiaire d'une mesure de similarité entre les champs de vitesse associés à ceux-ci. En effet, disposant de deux modèles de mouvements 2D, $\boldsymbol{\mu}_1$ et $\boldsymbol{\mu}_2$ localisés respectivement dans les régions $\mathcal{R}_1$ et $\mathcal{R}_1$ du plan, supposées (sans réduire la généralité du problème) centrées à l'origine du système de coordonnées, on peut définir une mesure de similarité entre ces deux modèles par :

$$s(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2) = \sum_{\mathbf{x} \in \mathcal{R}_1 \cup \mathcal{R}_2} \left\| \mathbf{v}_{\boldsymbol{\mu}_1}(\mathbf{x}) - \mathbf{v}_{\boldsymbol{\mu}_2}(\mathbf{x}) \right\| \ ,$$

où $\mathbf{v}_{\boldsymbol{\mu}_1}$ et $\mathbf{v}_{\boldsymbol{\mu}_2}$ représentent les champs générés à base des modèles $\boldsymbol{\mu}_1$ et $\boldsymbol{\mu}_2$, respectivement, et $\| \cdot \|$ est une norme usuelle dans $\mathbf{R}^2$. Cette définition ne peut pas être appliquée sous cette

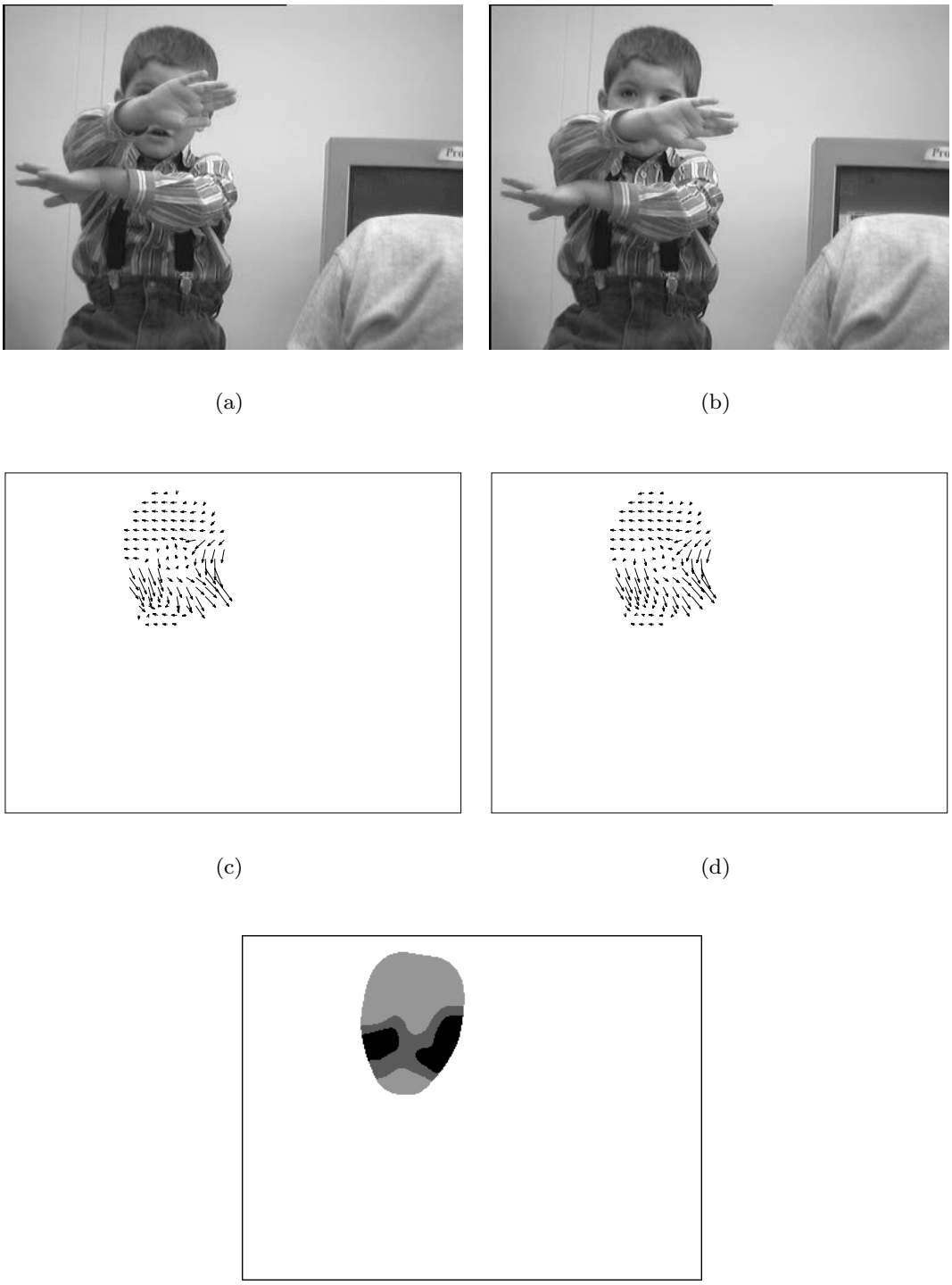
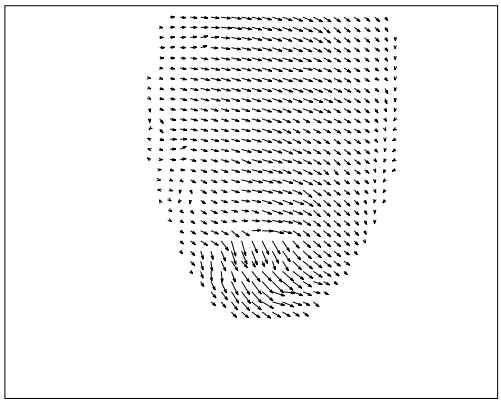


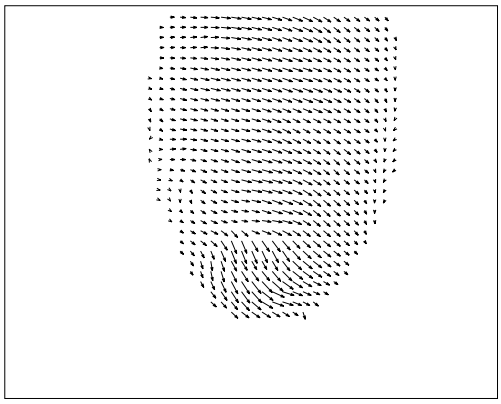
Figure 3.7 : Classification au sens du mouvement en présence d'occultations du visage : (a) et (b) images de test ; (c) champ de déplacements estimé dans la région du visage ; champ de déplacements régularisé par modélisation localement affine ; (d) résultat de la classification.

(a)

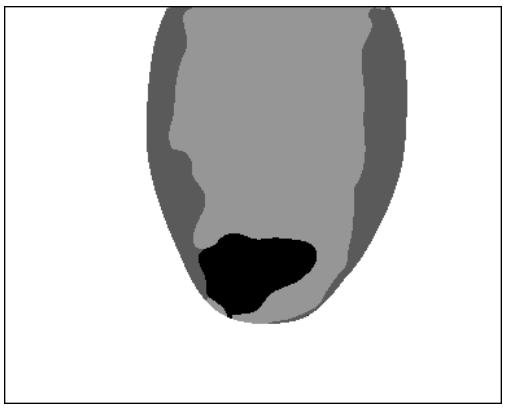
(b)



(c)



(d)



(e)

Figure 3.8 : Classification au sens du mouvement pour un visage parlant : (a) et (b) images de test ; (c) champ de déplacements estimé dans la région du visage ; (d) champ de déplacements régularisé par modélisation localement affine ; (e) résultat de la classification.

forme dans le cas des modèles affines dont nous disposons, car ils sont locaux. Par conséquent, nous l'adaptons à notre formalisme de la manière suivante. Soit $\delta(\mathbf{m}_i, \mathcal{C}_j)$ l'écart entre le modèle affine de $\mathbf{x}_i$ et la classe j, de moyenne $\boldsymbol{\mu}_j$, défini par :

$$\delta(\mathbf{m}_i, \mathcal{C}_j) = (1 - a_{ij}) \left\| \mathbf{v}_{\mathbf{m}_i}(\mathbf{x}_i) - \mathbf{v}_{\boldsymbol{\mu}_j}(\mathbf{x}_i) \right\| ,$$

où a_{ij} représente le coefficient d'appartenance de $\mathbf{x}_i$ à la classe $\mathcal{C}_j$. On définit l'ecart total entre deux classes $\mathcal{C}_j$ et $\mathcal{C}_k$, en sommant $\delta(\mathbf{m}_i, \mathcal{C}_j)$ sur tous les éléments de $\mathcal{C}_k$ et $\delta(\mathbf{m}_i, \mathcal{C}_k)$ sur toutes les éléments de $\mathcal{C}_j$:

$$\Delta(\mathcal{C}_j, \mathcal{C}_k) = \frac{\sum_{i=1}^{N} a_{ik}\, \delta(\mathbf{m}_i, \mathcal{C}_j)}{\sum_{i=1}^{N} a_{ik}} + \frac{\sum_{i=1}^{N} a_{ij}\, \delta(\mathbf{m}_i, \mathcal{C}_k)}{\sum_{i=1}^{N} a_{ij}} .$$

Evidemment, $\Delta(\mathcal{C}_j, \mathcal{C}_k) = \Delta(\mathcal{C}_k, \mathcal{C}_j)$.

Le cas pratique envisagé correspond à $k = 2$ classes (objet occultant/occulté). Dans cette situation, un critère simple, à base de seuil, sera utilisé pour décider si $\mathcal{C}_1$et $\mathcal{C}_2$ sont similaires en termes de mouvement et donc si l'occultation au sens du mouvement est présente ou non. Dans le cas où est pris le verdict de présence d'occultation est prise, la classe avec le plus grand nombre d'éléments sera désignée comme classe occultée (dans notre cas le visage).

3.5 Méthode robuste d'estimation de la pose 3D et résultats

En résumé, la procédure robuste d'estimation de la pose 3D du visage est schématisée Figure 3.9.

Pour évaluer objectivement les démarches adoptées, nous avons repris les simulations numériques sur les trois séquences calibrées, présentées dans la section 2.3.5 et les fonctions de distributions des erreurs ont été réévaluées. Les résultats, présentés Figure 3.10, montrent une amélioration sensible de la précision d'estimation pour la quasi-totalité des paramètres de pose 3D. Pour la séquence comportant des mouvements de vitesse modérée, dans 90% des cas la rotation dans le plan de l'image est estimée à 1° près au plus, tandis que les angles d'azimut et d'élévation le sont respectivement à 2° et 4° près au plus. Les translations sont estimées à 2% près au plus de la taille de la tête et le facteur d'échelle est calculé avec une précision relative supérieure à 97%.

Pour les deux autres séquences, comportant des mouvements rapides et très rapides, la rotation dans le plan de l'image est estimée à 1.5° près au plus, et l'azimut à 4° près au plus,

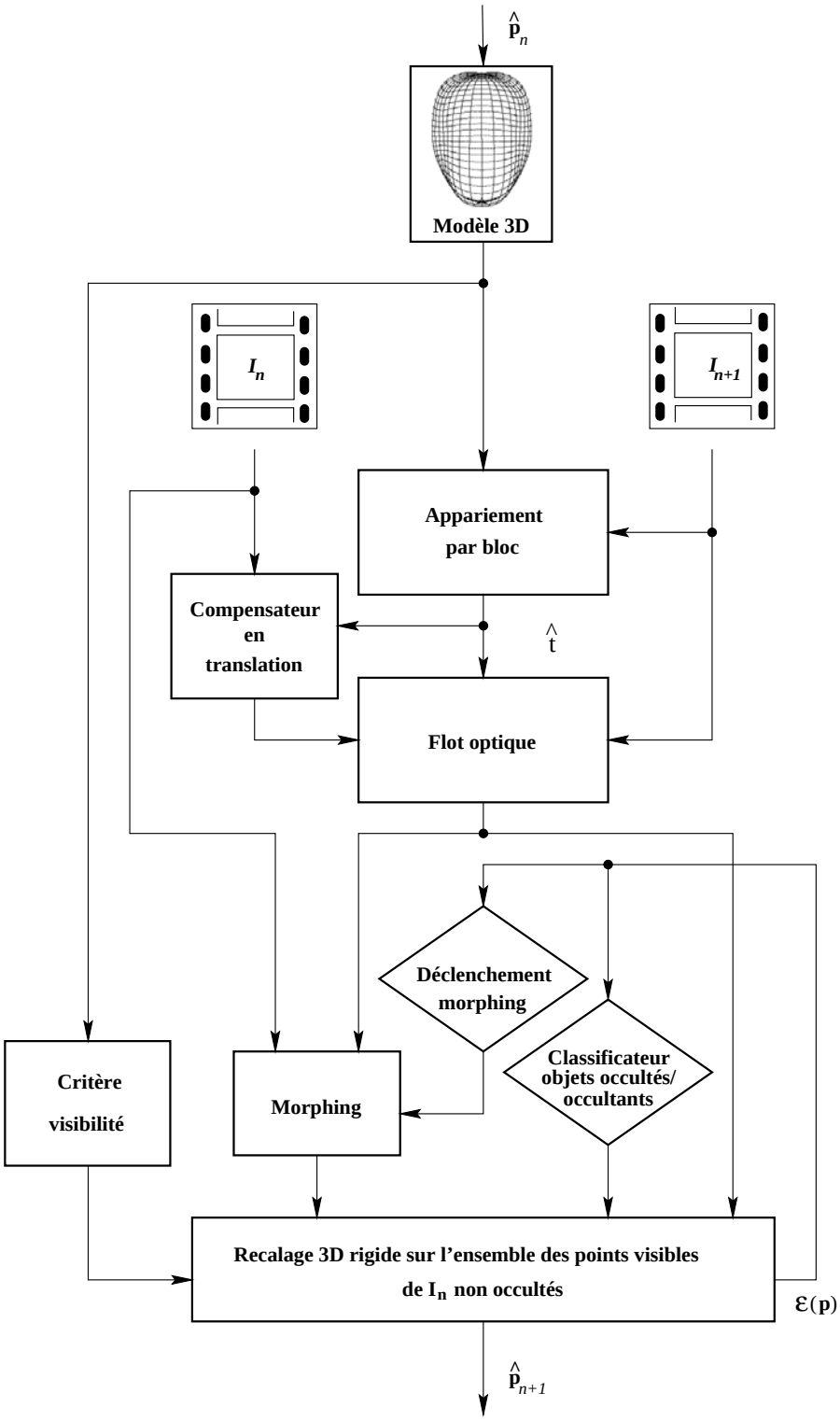


Figure 3.9 : Schéma synoptique de la méthode robuste d'estimation de la pose 3D.

dans 90% des cas. Les erreurs des translations ne dépassent pas 2% de la taille de la tête et le facteur d'échelle est estimé à 97% de sa valeur exacte. Toutefois, l'estimation de l'angle d'élévation n'est pas significativement améliorée.

Appliquée aux séquences réelles, la méthode proposée a permis de supprimer les échecs de recalage précédemment rencontrés (Figure 3.12) tout en en conservant les bonnes performances pour les configurations simples.

3.6 Conclusion

Dans ce chapitre, nous avons proposé des apports méthodologiques permettant de s'affranchir des limitations de la méthode d'estimation de la pose 3D de la tête à base de flot coloré. Il s'agit ici des décrochements survenus en présence de mouvements complexes de grande vitesse angulaire générant des auto-occultations, ou des occultations occasionnelles.

Utilisant le champ de déplacements compensé en translation dominante, nous avons développé une technique d'interpolation spatio-temporelle fondée sur une modélisation ondulatoire hiérarchique dans le cadre d'une approche physique par groupe de paquets d'onde. Cette interpolation nous a permis de synthétiser des images virtuelles afin de guider un recalage 3D/2D stable et précis en présence de grandes rotations. De plus, pour mieux gérer les auto-occultations de la tête, nous avons introduit un critère de visibilité issu des principes d'estimation robuste.

Ensuite, nous avons montré comment les cas d'occultation au sens du mouvement peuvent être gérés en considérant une classification fondée sur un critère de mouvement sous contrainte de régularité spatiale, couplée avec une analyse de similarité de mouvements à base de modèle paramétrique.

Les simulations numériques sur les séquences calibrées de synthèse montrent que l'estimation est effectuée dans 90% des cas avec une précision de 1.5° pour la rotation de la tête dans le plan de l'image, de 4° pour l'angle d'azimut, de 8° pour l'angle d'élévation, de 98% (relativement à la taille de la tête) pour les translations et de 97% pour le facteur d'échelle.

Appliquée aux séquences réelles, la méthode proposée a permis de supprimer les échecs de recalage précédemment rencontrés, tout en en conservant les bonnes performances pour les configurations simples.

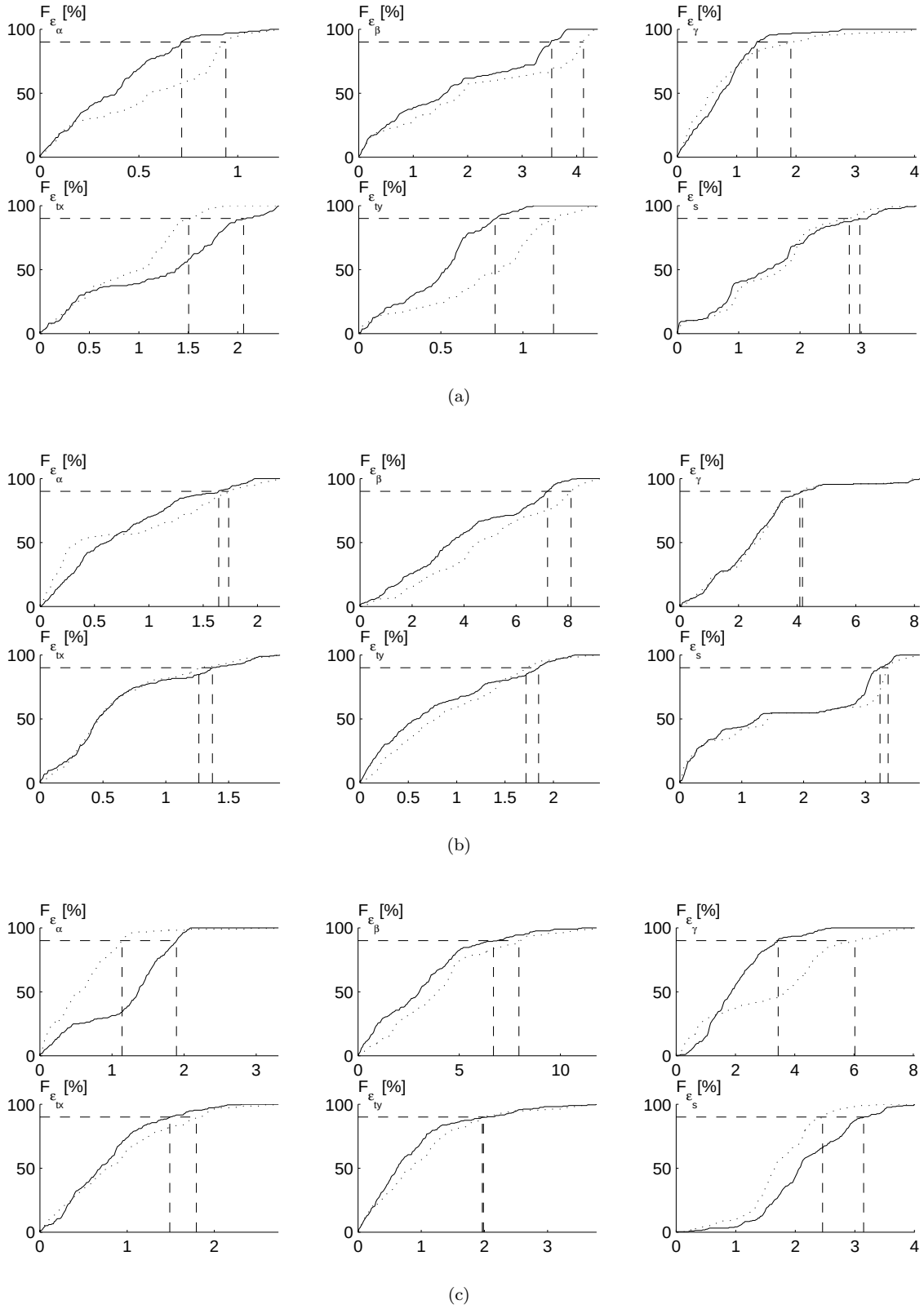


Figure 3.10 : Fonctions de distributions des erreurs d'estimation robuste pour les six paramètres de la pose 3D correspondant aux séquences synthétiques "Sorin" (a), "Corneliu" (b) et Armel (c).

Figure 3.11 : Résultats obtenus par la méthode robuste : (a) et (b) pour des rotations, (c) et (d) pour des occultations.

(a)

(b)

Figure 3.12 : La méthode robuste par flot coloré : suivi 3D effectué avec succès sur les séquences "Cornéliu" (a) et "Armel" (b).

Chapitre 4

Suivi de primitives de visage

4.1 Positionnement du problème

Recalage de primitives de visage et caractérisation d'expressions faciales à partir de séquences vidéos ont suscité ces dernières années de nombreuses recherches dans le cadre d'applications référencées vision telles que l'interaction homme-machine [Crowley95, Kobayashi97, Zelinsky96], la reconnaissance de visage ou d'expression faciale [Brunelli93, Chellappa95, Essa97] et le codage vidéo orienté modèle [Aizawa95, Bozdagi94, Eisert98, Li93, Reinders95, Zhang.L98]. Si le système visuel humain permet de localiser spontanément un visage et ses principales composantes, et de reconnaître et différencier les expressions faciales, ces mêmes tâches transposées dans le cadre de la vision par ordinateur restent des sujets ouverts à la recherche.

Les principales difficultés à surmonter renvoient à la grande variabilité morphologique du visage et aux déformations locales plus ou moins prononcées liées à la richesse des expressions faciales. En outre, les scènes à analyser, de contenus quelconques, sont acquises par une seule caméra fixe ou mobile, en général non calibrée et dans des conditions d'éclairement *a priori* inconnues et surtout non stabilisées.

Ciblant des applications de type codage vidéo à base de modèle, ce chapitre traite du suivi d'éléments faciaux tels que les yeux et la bouche dans des séquences vidéos, à l'aide de modèles déformables compatibles avec le nouveau standard MPEG-4-SNHC (*Synthetic and Natural Hybrid Coding*) [MPEG-4].

Tout d'abord, nous présentons et détaillons le principe du recalage à base de prototypes déformables et les outils mathématiques dont nous avons besoin pour mettre en œuvre un ajustement précis et une caractérisation efficace des régions de l'image correspondant aux yeux et à la bouche. Dans la suite, nous décrivons la modélisation de ces éléments faciaux par prototypes

déformables adaptés aux descripteurs de visage définis par le standard MPEG-4. Typiquement, les modèles de bouche et d'œil sont obtenus par interpolation des paramètres MPEG-4 de définition du visage à l'aide de B-splines. Ils sont intrinsèquement caractérisés par des contraintes élastiques et de symétrie locale héritées d'une modélisation physique à base de ressorts. Dans le cas particulier de l'œil, deux prototypes sont proposés, correspondant respectivement à la configuration ouverte/fermée de celui-ci.

Ensuite, nous spécifions les déformations auxquelles sont soumis les prototypes par interaction avec les données images. Ici, nous introduisons des primitives de gradient et de texture et les combinons à une carte de segmentation floue sous contrainte spatiale, de manière à exploiter les caractéristiques de chaque élément facial considéré. Les fonctionnelles d'énergie externe sont définies et justifiées.

Nous poursuivons en énonçant le principe du suivi des éléments faciaux et en détaillant l'algorithme associé. Les contraintes internes et externes des prototypes déformables sont combinées dans un schéma d'optimisation par la méthode du simplexe. L'initialisation robuste des prototypes d'une image à l'autre est effectuée en couplant une procédure de segmentation automatique de l'iris et de détection de la configuration ouverte/fermée de l'œil à l'estimation de la pose 3D de la tête décrite dans le chapitre précédent. La stabilité et la précision des résultats sont établies à partir de séquences vidéos de visage parlant, acquises dans des conditions réalistes.

Enfin, l'algorithme de suivi de primitives de visage est intégré dans un schéma d'analyse/synthèse de déformations faciales, compatible MPEG-4, pour l'animation d'avatars à partir de séquences vidéos naturelles.

4.2 Principe du recalage par prototypes déformables et outils associés

4.2.1 Prototypes déformables

Introduits par Yuile et al. [Yuille92] dans le cadre d'applications d'extraction de primitives de visage, les prototypes déformables offrent la souplesse nécessaire à une représentation fiable et en même temps suffisamment variable des déformations locales de visage.

De manière générale, les prototypes déformables comportent trois éléments de base :

- un modèle géométrique paramétré associé à la primitive de visage ciblée, exploitant des connaissances *a priori* sur la forme de celle-ci,

- un modèle d'interaction avec l'image, spécifiant la correspondance entre les éléments de géométrie du prototype et un ensemble de primitives d'image, et
- un algorithme de recalage du prototype sur l'image, capable de s'affranchir des problèmes de variation d'échelle, inclinaison ou rotation de la tête et des conditions d'éclairement.

A partir de l'ensemble de paramètres le définissant, le modèle géométrique associé au prototype doit assurer une représentation fiable de la forme de la primitive de visage correspondante et prendre en compte sa variabilité en présence de déformations. Pour satisfaire à ces demandes, notre approche définit le modèle géométrique par un ensemble de courbes paramétriques générées par des splines, dont le formalisme est décrit dans la section 4.2.2.

L'interaction entre le prototype et l'image est réalisée à travers une fonction d'énergie qui formalise un ensemble de descriptions qualitatives liées à la primitive de visage ciblée (par exemple, l'œil se caractérise par la présence d'une région foncée - l'iris - entourée par le blanc de l'œil). Dans l'approche originelle, les composantes de la fonction d'énergie, définies en termes de "dénivellation" topographique (dômes et vallées), intensité d'image, gradients et contraintes internes du modèle géométrique, font évoluer ce-dernier d'une position initiale à la position recalée. Les représentations de type dômes, vallées, contours sur lesquelles le prototype agit, sont obtenues à partir de l'image d'origine par différents filtrages (morphologiques, gradients, ...). Pour assurer des interactions d'une plus grande portée, ces filtrages sont généralement suivis d'un lissage avec un noyau de type $\gamma(x,y) = e^{-\rho(x^2+y^2)^{1/2}}$ permettant ainsi l'élargissement de l'effet d'attraction du modèle vers la position d'équilibre. Si Ψ_{d}, Ψ_{v}, Ψ_{c} désignent respectivement les images segmentées de dômes, vallées et contours, les représentations utilisées dans l'expression de la fonction d'énergie s'expriment par :

$$\begin{aligned} \Phi_{\mathrm{d}} &= \gamma * \Psi_{\mathrm{d}} \\ \Phi_{\mathrm{v}} &= \gamma * \Psi_{\mathrm{v}} \ , \\ \Phi_{\mathrm{c}} &= \gamma * \Psi_{\mathrm{c}} \end{aligned}$$

où $(\cdot * \cdot)$ est l'opérateur de convolution.

Dans notre approche, les régions d'intérêt du visage sont extraites lors de l'étape de prétraitement par segmentation floue sous contrainte spatiale, dont le formalisme est présenté dans la section 4.2.3.

La fonction d'énergie s'exprime comme une combinaison de plusieurs composantes :

$$E = E_{\mathrm{d}} + E_{\mathrm{v}} + E_{\mathrm{c}} + E_{\mathrm{i}} + E_{\mathrm{int}} \ ,$$

chacune d'entre elles étant définie sous forme d'une fonction des paramètres du prototype et/ou de la représentation Φ associée et présentant un minimum global pour les valeurs des paramètres

correspondant à la position recalée du prototype. Ici, E_{d}, E_{v}, E_{c} désignent respectivement les énergies associées aux représentations "dômes", "vallées" et "contours", E_{i} une énergie qui exploite l'information de l'image originale et E_{int} l'énergie interne associée au modèle géométrique du prototype.

E_{d}, E_{v} et E_{i} sont généralement définies sous forme d'intégrales de surface sur les régions pertinentes du modèle :

$$\begin{aligned} E_{\mathrm{d}} &= \frac{k_{\mathrm{d}}}{|S_{\mathrm{d}}|} \int_{S_{\mathrm{d}}} \Phi_{\mathrm{d}} \, dS \\ E_{\mathrm{v}} &= \frac{k_{\mathrm{v}}}{|S_{\mathrm{v}}|} \int_{S_{\mathrm{v}}} \Phi_{\mathrm{v}} \, dS \ , \\ E_{\mathrm{i}} &= \frac{k_{\mathrm{i}}}{|S_{\mathrm{i}}|} \int_{S_{\mathrm{i}}} \Phi_{\mathrm{i}} \, dS \end{aligned}$$

où k_{d}, k_{v}, k_{i} sont des coefficients de pondération et S_{d}, S_{v}, S_{i} les régions pertinentes du modèle associées à chaque représentation Φ.

De manière similaire, E_{c} s'exprime par une intégrale le long des contours représentatifs du modèle :

$$E_{\mathrm{c}} = \frac{k_{\mathrm{c}}}{|L_{\mathrm{c}}|} \int_{L_{\mathrm{c}}} \Phi_{\mathrm{c}} \, dl \ .$$

L'énergie interne E_{int} est étroitement liée au modèle géométrique du prototype et définit les interactions entre ses différentes composantes, sous forme de position relative et contraintes d'élasticité.

Le recalage du prototype sur l'image revient à minimiser l'énergie E sur l'espace des paramètres du modèle.

4.2.2 Modélisation géométrique à base de splines

La modélisation par splines est une technique de génération d'objets géométriques réguliers, largement répandue dans le monde du graphique par ordinateur [Bartels87]. Privilégiant des représentations paramétriques compactes pour des objets relativement complexes tout en gardant la précision d'approximation, les splines ont également été utilisées avec succès dans le cadre de l'imagerie faciale [Eisert98, Moses95, Sanchez97, Terzopoulos93]. Le formalisme de la modélisation à base de fonctions splines est présenté dans cette section.

4.2.2.1 Généralités

Introduisons tout d'abord la notion de fonction spline sur un ensemble de points réels.

Définition 4.1 (spline de degré m) Soit $t_0 \leq t_1 \leq \ldots \leq t_n$ un ensemble de nombres réels (nommés nœuds). Une fonction s est une spline de degré m sur $\mathbf{t} = (t_0, t_1, \ldots, t_n)$ si :

1° s est un polynôme de degré $d \leq m$ dans chaque intervalle

$$(-\infty, t_0), [t_0, t_1), \ldots, [t_{n-1}, t_n], (t_n, \infty) ;$$

2° s est de classe $C^{(m-1)}$ en chaque nœud t_i *i.e.*

$$s^{(p)}(t_{i-}) = s^{(p)}(t_i) = s^{(p)}(t_{i+}), \begin{cases} p = 0, 1, \ldots, m-1 \\ i = 0, 1, \ldots, n \end{cases} .$$

L'ensemble des splines de degré m sur $\mathbf{t}$ sera noté par $\mathcal{S}_{m,\mathbf{t}}$.

□

Evidemment, tout polynôme de degré m est une spline de degré m, mais, dans le cas général, les splines sont des fonctions polynomiales par morceaux. Par exemple, une spline de degré 3 (spline cubique) consiste en une concaténation de fragments cubiques telle que la courbe résultante est continue, et de pentes et courbures continues aux points de raccordement.

Une classe particulière de splines, nommées naturelles, présente un intérêt spécial en pratique.

Définition 4.2 (spline naturelle de degré m) Une spline $s \in \mathcal{S}_{m,\mathbf{t}}$ est dite naturelle si :

1° $m = 2k - 1$, $k \in \mathbf{N}^*$;

2° s est un polynôme de degré $d \leq k - 1$ sur les intervalles $(-\infty, t_0)$ et (t_n, ∞).

L'ensemble des splines naturelles de degré m sur $t_0, t_1, \ldots, t_n$ sera noté par $\widetilde{\mathcal{S}}_{m,\mathbf{t}}$.

□

Comme une conséquence de la continuité des dérivées de s, remarquons que $s \in \widetilde{\mathcal{S}}_{2k-1,\mathbf{t}}$ implique :

$$s^{(p)}(t_0) = s^{(p)}(t_n), \text{ pour tout } p = k, k+1, \ldots, 2k-2. \tag{4.1}$$

Une spline cubique naturelle, par exemple, est de courbure nulle en ses nœuds terminaux.

En pratique, le comportement de la spline à l'extérieur de l'intervalle $[t_0,\, t_n]$ présente peu d'intérêt. Considérons, par conséquent, la restriction d'une spline $s \in \mathcal{S}_{m,\,\mathbf{t}}$ à l'intervalle $[t_0,\, t_n)$:

$$s(t) = \sum_{i=1}^{n} \phi_{[t_{i-1},\, t_i)} \left[c_{0,i} + c_{1,i}\,(t - t_{i-1}) + \ldots + c_{m,i}\,(t - t_{i-1})^m \right] \ ,$$

où $\phi_{\mathcal{I}}$ est la fonction caractéristique de l'intervalle $\mathcal{I}$. Les contraintes de raccordement aux points nodaux intérieurs à l'intervalle d'intérêt, $t_1,\, \ldots,\, t_{n-1}$, s'expriment sous la forme :

$$\begin{array}{lcl} \displaystyle\sum_{j=0}^{m} c_{j,i}\,(t_i - t_{i-1})^j & = & c_{0,i+1} \\ \displaystyle\sum_{j=1}^{m} c_{j,i}\, j\,(t_i - t_{i-1})^{j-1} & = & c_{1,i+1} \\ \vdots & & \\ \displaystyle\sum_{j=m-1}^{m} c_{j,i}\,\frac{j!}{(m-1)!}\,(t_i - t_{i-1})^{j-(m-1)} & = & c_{m-1,i+1} \end{array} \qquad i = 1,\, \ldots,\, n-1 \tag{4.2}$$

i.e. $m(n-1)$ contraintes associées à $(m+1)n$ coefficients $c_{j,i}$. En conséquence, la spline s dispose de

$$\nu = (m+1)\,n - m\,(n-1) = n + m$$

degrés de liberté. Dans le cas d'une spline naturelle, le nombre de contraintes augmente de $2(k-1) = m-1$, donc le nombre de degrés de liberté devient :

$$\widetilde{\nu} = (n-m) - (m-1) = n + 1 \ .$$

En conséquence, une spline naturelle de degré donné, interpolant un ensemble de points dont les abscisses sont les t_i, $i = 0,\, \ldots,\, n$ est déterminée de manière unique. Dans ce cas, les coefficients $c_{j,i}$ peuvent être exprimés directement, comme la solution du système linéaire formé par les équations (4.1) et (4.2). Une autre approche, permettant l'interpolation dans le cas où les abscisses ne coïncident pas avec les t_i, fait appel à la notion de spline de base (abrégé B-spline) et sera traitée dans la suite.

4.2.2.2 B-splines

Les $n+m$ degrés de liberté d'une spline arbitraire $s \in \mathcal{S}_{m,\,\mathbf{t}}$ nous indique l'existence d'une certaine structure d'espace vectoriel de dimension $n+m$ sur $\mathcal{S}_{m,\,\mathbf{t}}$ [DeBoor78]. Une base $\left\{\beta_{i,\,m,\,\mathbf{t}}\right\}_{i=0,\,\ldots,\,n+m-1}$ de cet espace, dont la construction est présentée ci-dessous, définit les B-splines associées avec m et $\mathbf{t}$.

Définition 4.3 (B-splines) Soit $t_{-m} \leq t_{-m+1} \leq \ldots \leq t_0 \leq \ldots \leq t_n \leq \ldots \leq t_{n+m-1} \leq t_{n+m}$ un ensemble augmenté de nœuds associé à $\mathbf{t} = (t_0, t_1, \ldots, t_n)$. Les fonctions $\beta_{i,m,\mathbf{t}}$ définies récursivement par :

$$\begin{aligned} \beta_{i,0,\mathbf{t}}(t) &= \begin{cases} 1\,, & \text{si } t_i < t < t_{i+1} \\ 0\,, & \text{sion} \end{cases} \\ \beta_{i,k,\mathbf{t}}(t) &= \omega_{i,k}(t)\,\beta_{i,k-1,\mathbf{t}}(t) + [1 - \omega_{i+1,k}(t)]\,\beta_{i+1,k-1,\mathbf{t}}(t)\ ,\ k = 1, \ldots, m \end{aligned} \tag{4.3}$$

avec

$$\omega_{i,k}(t) = \begin{cases} \dfrac{t - t_{i-k}}{t_i - t_{i-k}}\,, & \text{si } t_i \neq t_{i-k} \\ 0\,, & \text{sion} \end{cases}\ ,$$

s'appellent B-splines de degré m sur $\mathbf{t}$. $t_0, t_1, \ldots, t_n$ définissent les nœuds internes et les nœuds ajoutés s'appellent des nœuds externes.

□

Théorème 4.1 Soient $\mathbf{t}$ et m fixés. Alors :

1° les fonctions $\beta_{i,m,\mathbf{t}}$ précédemment définies sont splines d'ordre m sur $\mathbf{t}$ et elles satisfont :

$$\beta_{i,m,\mathbf{t}}(t) \quad \begin{cases} > 0\,, & \text{si } t_{i-m} < t < t_{i+1} \\ = 0\,, & \text{sion} \end{cases}\ ;$$

2° les $\beta_{i,m,\mathbf{t}}$ forment une partition de l'unité sur l'intervalle $[t_0, t_n]$, *i.e.*

$$\sum_{i=0}^{n+m-1} \beta_{i,m,\mathbf{t}}(t) = 1 \ \textit{pour}\ t_0 < t < t_n$$

3° les B-splines $\beta_{i,m,\mathbf{t}}$ permettent d'exprimer $\mathcal{S}_{m,\mathbf{t}}$ comme :

$$\mathcal{S}_{m,\mathbf{t}} = \left\{ \sum_{i=0}^{n+m-1} a_i\,\beta_{i,m,\mathbf{t}} \mid a_0, \ldots, a_{n+m-1} \in \mathbf{R} \right\}\ .$$

□

La Figure 4.1 montre les B-splines asociées à un ensemble de nœuds internes uniformément répartis dans l'intervalle [0, 1] et de nœuds externes coïncidant avec les extrémités.

Le problème de l'interpolation à base de B-splines consiste à établir les hypothèses dans lesquelles, si on se donne un ensemble de couples (x_i, y_i) et un degré m, il existe un vecteur de nœuds $\mathbf{t}$ et une spline $s = \sum_i a_i\,\beta_{i,m,\mathbf{t}} \in \mathcal{S}_{m,\mathbf{t}}$ tels que $s(x_i) = y_i$, pour tout i. Les conditions d'existence et d'unicité de cet interpolateur sont données par le théorème suivant.

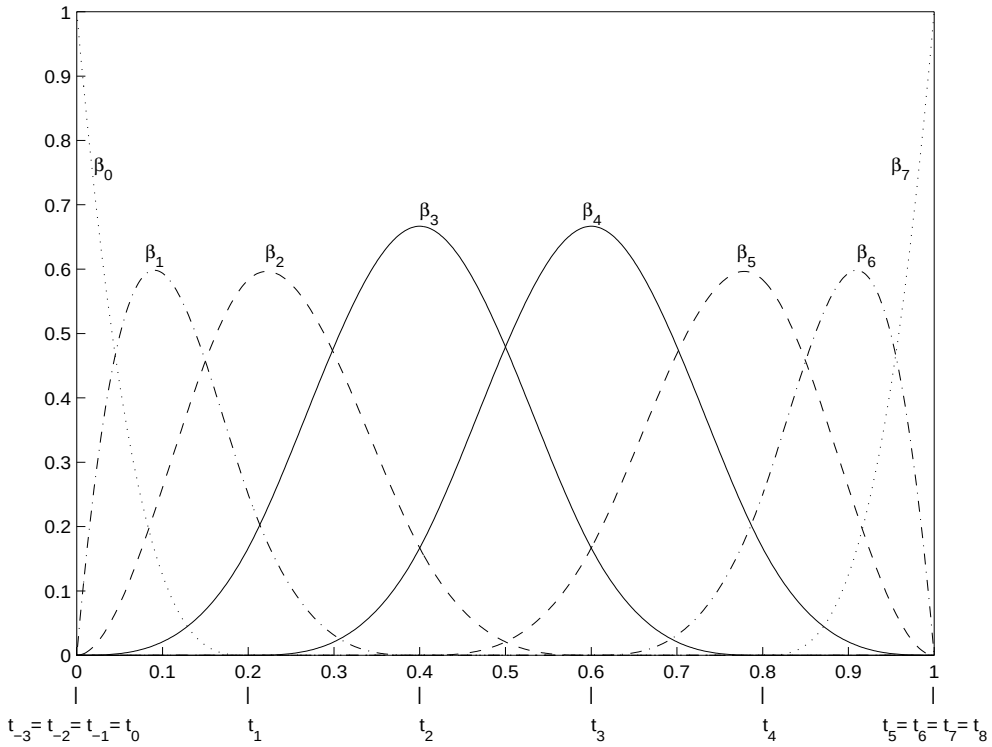


Figure 4.1 : Les 8 B-splines cubiques résultés de l'ensemble de nœuds internes $t_0 = 0$, $t_1 = 0.2$, $t_2 = 0.4$, $t_3 = 0.6$, $t_4 = 0.8$, $t_5 = 1.0$ et de nœuds externes $t_{-3} = t_{-2} = t_{-1} = t_0$, $t_8 = t_7 = t_6 = t_5$.

Théorème 4.2 (Schoenberg-Whitney) Soient n et m deux entiers positifs fixés et soit $(x_i,\, y_i)$, $i = 1,\, \ldots,\, n+m$ un ensemble donné de couples réelles. Un ensemble de nœuds $t_0 \leq t_1 \leq \ldots \leq t_n$ défini une spline unique $s \in \mathcal{S}_{m,\mathbf{t}}$ telle que $s(x_i) = y_i$, $i = 1,\, \ldots,\, n+m$, si seulement si :

1° $t_0 \leq x_1 < x_2 < \ldots < x_{n+m} \leq t_n$;

2° $x_i < t_i < x_{i+m+1}$, $i = 1,\, \ldots,\, n-1$ (la condition de Schoenberg-Whitney).

□

Observations :

1. la condition de Schoenberg-Whitney impose l'existence d'au moins un point d'interpolation $x_k \in [t_0,\, t_1)$ et de même au moins un point $x_l \in (t_{n-1},\, t_n]$;
2. pour $1 \leq i < n-2$ il est permis d'avoir des intervalles $[t_i,\, t_{i+1}]$ sans points d'interpolation ; au plus m intervalles consécutifs de ce type peuvent exister.

L'interpolateur s cherché sera de la forme :

$$s(t) = \sum_{i=0}^{n+m-1} a_i\, \beta_{i,\,m,\,\mathbf{t}}(t)\ . \tag{4.4}$$

Son calcul nécessite de déterminer tout d'abord les splines de base $\beta_{i,\,m,\,\mathbf{t}}$. Il s'agit d'introduire $2\,m$ nœuds supplémentaires, $t_{-m},\,\ldots,\,t_{-1},\,t_{n+1},\,\ldots,\,t_{n+m}$ et d'appliquer les formules (4.3). $\beta_{i,\,m,\,\mathbf{t}}$ étant disponibles, l'équation (4.4) particularisée pour chaque couple $(x_j,\,y_j)$ conduit au système linéaire :

$$\sum_{i=0}^{n+m-1} a_i\,\beta_{i,\,m,\,\mathbf{t}}(x_j) = y_j,\;\; j = 1,\,\ldots,\,n+m\;.$$

La matrice de ce système est de type diagonale par bloc suite au fait que les fonctions $\beta_{i,\,m,\,\mathbf{t}}$ sont de support fini :

$$A = \begin{pmatrix} \boxed{} & & & \\ & \boxed{} & & \\ & & \ddots & \\ & & & \boxed{} \end{pmatrix}.$$

Ainsi, des méthodes spécifiques peuvent être mises en œuvre pour le résoudre. L'élimination gaussienne, par exemple, est stable pour ce type de système [Cox77]. Remarquons que même si la solution du système et les B-splines $\beta_{i,\,m,\,\mathbf{t}}$ dépendent du choix des nœuds ajoutés, la spline résultante s n'en dépend pas, puisque elle est unique, selon le théorème de Schoenberg-Whitney.

4.2.3 Segmentation floue sous contrainte spatiale

La segmentation automatique d'images constitue un problème typique de classification de données sous contraintes spatiales. Le but consiste à établir un ensemble de parties 2D tel que chaque partie regroupe des valeurs observées de l'image – unidimensionnelles ou multidimensionnelles – voisines aussi bien dans l'espace que du point de vue d'un critère d'homogénéité. Par conséquent, une procédure de segmentation automatique devrait répondre aux exigences suivantes :

- que chaque partie regroupe des valeurs aussi proches que possible au sens d'une mesure de ressemblance ;
- que le support de chaque partie satisfasse certaines contraintes de régularité, en postulant, par exemple, que les pixels ont d'autant plus de chance d'appartenir à la même partie qu'ils sont plus proches spatialement.

Dans le cadre de l'imagerie faciale, la segmentation de la région du visage et à l'intérieur de cette région est largement utilisée en tant qu'étape de pré-traitement. Particulièrement performantes en termes de résultats, les techniques de segmentation mettant en œuvre la modélisation

de l'information chromatique sous forme de mélange de distributions ont été appliquées avec succès pour la détection du visage [Moghaddam97, Yang99], ou pour le suivi de primitive faciale [Tian99, Torre2000].

Dans ce contexte, nous présentons ici les principales notions et approches algorithmiques liées à la classification automatique de données spatiales dans un contexte de modélisation probabiliste de type mélange de distributions.

4.2.3.1 Modèle de mélange

Dans le cadre de la classification automatique, on cherche à déterminer, à partir d'un ensemble d'observations $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\} \subset \mathbf{R}^d$, un regroupement en K composantes, chacune caractérisée par sa propre densité de probabilité, $p_k(\mathbf{x})$. Le modèle de distributions mélangées constitue donc un formalisme bien adapté à ce cadre. Ce type de modèle suppose que les observations $\mathbf{x}_1, \ldots, \mathbf{x}_N$ sont des réalisations de N vecteurs aléatoires, $\mathbf{X}_1, \ldots, \mathbf{X}_N$, indépendants dont la densité de probabilité jointe est modélisée comme une combinaison linéaire de densités de probabilité paramétriques de même type.

Définition 4.4 Soient $\mathbf{X}_1, \ldots, \mathbf{X}_N$, N vecteurs aléatoires réels d-dimensionnels définis sur le même espace de probabilité. Ces vecteurs suivent une distribution de mélange de K composantes sur un espace de paramètres $\mathcal{P}$ si :

1° $\mathbf{X}_1, \ldots, \mathbf{X}_N$ sont indépendants ;

2° la densité de probabilité jointe de $\mathbf{X}_1, \ldots, \mathbf{X}_N$ s'exprime sous la forme[1] :

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \sum_{k=1}^{K} \pi_k \, f(\mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}_k) \tag{4.5}$$

où

- $\boldsymbol{\theta}_k \in \mathcal{P}$, $k = 1, \ldots, K$ sont les paramètres des composantes du mélange ;
- $f : \mathbf{R}^{Nd} \times \mathcal{P} \to \mathbf{R}$ est une densité de probabilité par rapport aux variables $\mathbf{x}_1, \ldots, \mathbf{x}_N$ ($f \geq 0$ et $\int_{\mathbf{R}^n} f \, d\mathbf{x}_1 \ldots d\mathbf{x}_N = 1$), pour tout $k = 1, \ldots, K$;
- $0 < \pi_k < 1$, $k = 1, \ldots, K$, représentent les proportions des composantes du melange et $\sum_{k=1}^{K} \pi_k = 1$.

[1] Pour des raisons de simplicité, la notation de la densité de probabilité ne contient pas explicitement le nom du vecteur aléatoire auquel elle est associée, celui-ci résultant du contexte. Par exemple, $p(\mathbf{x}_i)$ sera la densité de $\mathbf{X}_i$ et $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ la densité de $(\mathbf{X}_1, \ldots, \mathbf{X}_N)$.

□

Intuitivement, $f(\mathbf{x}_1, \ldots, \mathbf{x}_N, \boldsymbol{\theta}_k)$ représente la densité de probabilité jointe de $\mathbf{X}_1, \ldots, \mathbf{X}_N$ conditionnée par la réalisation de la composante k du mélange. Elle est notée $p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta}_k)$. De même, $p(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ donné par (4.5) est interprétée en termes de densité de probabilité conditionnée par la réalisation d'un mélange de paramètres $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ et proportions $\pi_1, \ldots, \pi_K$; elle sera notée par $p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\Theta})$, où $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K, \pi_1, \ldots, \pi_K)$ désigne les paramètres globaux du mélange.

Les densités marginales du mélange sont données par :

$$\begin{aligned} p(\mathbf{x}_n \mid \boldsymbol{\Theta}) &= \int_{\mathbf{R}^{(N-1)d}} p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\Theta})\, d\mathbf{x}_1 \ldots d\mathbf{x}_{n-1}\, d\mathbf{x}_{n+1} \ldots d\mathbf{x}_N \\ &= \int_{\mathbf{R}^{(N-1)d}} d\mathbf{x}_1 \ldots d\mathbf{x}_{n-1}\, d\mathbf{x}_{n+1} \ldots d\mathbf{x}_N \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta}_k) \\ &= \sum_{k=1}^{K} \pi_k \int_{\mathbf{R}^{(N-1)d}} p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\theta}_k)\, d\mathbf{x}_1 \ldots d\mathbf{x}_{n-1}\, d\mathbf{x}_{n+1} \ldots d\mathbf{x}_N \\ &= \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \end{aligned} \tag{4.6}$$

i.e. chaque $\mathbf{X}_n$ suit une distribution de mélange de paramètres $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$ et de proportions $\pi_1, \ldots, \pi_K$ (les $\mathbf{X}_i$ sont identiquement distribués). Prenant en compte l'indépendance des $\mathbf{X}_n$ et (4.6), la densité de probabilité jointe s'exprime sous la forme :

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N \mid \boldsymbol{\Theta}) = \prod_{n=1}^{N} p(\mathbf{x}_n \mid \boldsymbol{\Theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k\, p(\mathbf{x}_n \mid \boldsymbol{\theta}_k)\ . \tag{4.7}$$

Exemple En l'absence de connaissance *a piori* concernant les distributions du mélange, la modélisation gaussienne multivariée est couramment utilisée. Dans ce cas, chaque composante k du mélange est complètement déterminée par sa valeur moyenne $\boldsymbol{\mu}_k \in \mathbf{R}^d$ et sa matrice de covariance $\boldsymbol{\Sigma}_k$ – symétrique et définie positive – *i.e.* $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ et les densités conditionnelles sont exprimées par :

$$p(\mathbf{x}_n \mid (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) = (2\pi)^{-d/2}\, [\det(\boldsymbol{\Sigma}_k)]^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{x}_n - \boldsymbol{\mu}_k) \cdot \boldsymbol{\Sigma}_k^{-1} \cdot (\mathbf{x}_n - \boldsymbol{\mu}_k)^t\right]\ .$$

Ici, $\mathbf{x}_n$ et $\boldsymbol{\mu}_k$ sont des vecteurs lignes et t désigne la transposée.

4.2.3.2 Estimation des paramètres du mélange

4.2.3.2.a Maximum de vraisemblance Déterminer les paramètres $\boldsymbol{\theta}_k$ et les proportions π_k d'un modèle de mélange à partir de données expérimentales constitue un problème typique d'estimation statistique de paramètres. Plus précisément, il s'agit d'établir des formules permettant le calcul de $\boldsymbol{\theta}_k$ et π_k en fonction d'une réalisation $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ de $(\mathbf{X}_1, \ldots, \mathbf{X}_N)$. L'une des méthodes d'estimation les plus utilisées vise à maximiser la fonction de vraisemblance définie par :

$$L(\boldsymbol{\Theta}) = p(\mathbf{x} \mid \boldsymbol{\Theta}) \ ,$$

ou par son logarithme[2] :

$$L(\boldsymbol{\Theta}) = \ln p(\mathbf{x} \mid \boldsymbol{\Theta}) \ .$$

La valeur

$$\widehat{\boldsymbol{\Theta}}_{\mathrm{mv}} = \arg\max_{\boldsymbol{\Phi}} \ (\ln)\, p(\mathbf{x} \mid \boldsymbol{\Theta}) \tag{4.8}$$

s'appelle estimé du maximum de vraisemblance.

L'approche la plus directe pour déterminer $\widehat{\boldsymbol{\Theta}}_{\mathrm{mv}}$ consiste à résoudre (4.8) par l'annulation des dérivées partielles de L sous contrainte que la matrice hessienne de L soit définie négative. Le recours à cette démarche aboutit au résultat dans le cas où les distributions sont modélisées par des expressions simples. En général, quand il s'agit de distributions non triviales, l'approche analytique n'est plus possible. Un exemple typique est celui d'une répartition de mélange. Utilisant (4.7), la log-vraisemblance s'exprime par :

$$L(\boldsymbol{\Theta}) = \sum_{n=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k \, p(\mathbf{x}_n \mid \boldsymbol{\theta}_k) \right] \ .$$

Cette expression n'est pas facile à traiter dans un contexte d'étude analytique puisqu'elle contient le logarithme d'une somme. Dans de telles situations il est nécessaire de faire appel à des techniques numériques d'optimisation. Une méthode bien adaptée à ce type de problème est l'algorithme EM[3], présenté dans la suite.

4.2.3.2.b L'algorithme EM L'observation selon laquelle les composantes $\mathbf{X}_n$ d'un mélange sont identiquement distribuées permet l'interprétation suivante : une réalisation $\mathbf{x}_n$ de $\mathbf{X}_n$ est générée par un tirage entre les K distributions $p(\mathbf{x}_n \mid \boldsymbol{\theta}_k)$, selon les probabilités $\pi_1, \ldots, \pi_K$, suivi d'un tirage à l'intérieur d'une classe (dans la suite, le terme classe sera également utilisé

[2] L'utilisation du logarithme conduit à des calculs simples dans le cas où $p(\mathbf{x} \mid \boldsymbol{\Phi})$ comporte des exponentielles comme, par exemple, la densité gaussienne.

[3] EM représente l'abréviation du nom anglais *Expectation Maximisation*.

pour désigner composante du mélange). Cela implique l'existence d'une classification cachée $\mathbf{Y} = (y_1, \ldots, y_N)$ dont les éléments $y_n \in \{1, \ldots, K\}$ sont statistiquement indépendants (suite à l'indépendance des $\mathbf{X}_n$). Les probabilités des classes étant $\pi_1, \ldots, \pi_K$, il résulte que $\mathbf{Y}$ suit une distribution multinomiale de paramètre $(\pi_1, \ldots, \pi_K)$.

Dans ce contexte, $\mathbf{x}$ seul est traité en tant qu'observation incomplète, tandis que $\mathbf{y}$, la réalisation $\mathbf{Y}$, représente l'information manquante. Le vecteur joint $(\mathbf{x}, \mathbf{y})$ représente les données complétées sur lesquelles l'estimation des paramètres sera effectuée. Le principe de maximisation de la vraisemblance sera donc appliqué à la densité de probabilité jointe[4] :

$$p(\mathbf{x},\ \mathbf{y} \mid \boldsymbol{\Theta}) = p(\mathbf{x} \mid \mathbf{y},\ \boldsymbol{\Theta})\, P(\mathbf{y} \mid \boldsymbol{\Theta})\ . \tag{4.9}$$

$p(\mathbf{x} \mid \mathbf{y},\ \boldsymbol{\Theta})$ représente la densité de probabilité de $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ dans l'hypothèse où on a observé la classification $(y_1, \ldots, y_N)$ et les paramètres $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K$, c'est-à-dire $\mathbf{x}_n$ provient de la classe y_n de paramètre $\boldsymbol{\theta}_{y_n}$, pour $n = 1, \ldots, N$. De même, $P(\mathbf{y} \mid \boldsymbol{\Theta})$ est la probabilité de $(y_1, \ldots, y_N)$ dans l'hypothèse où les classes ont les proportions $\pi_1, \ldots, \pi_K$. Ainsi, la log-vraisemblance des données complétées sera donnée par :

$$\begin{aligned} L_{\mathrm{c}}(\boldsymbol{\Theta}) &= \ln p\,(\mathbf{x},\ \mathbf{y} \mid \boldsymbol{\Theta}) = \ln\left[p(\mathbf{x} \mid \mathbf{y},\ \boldsymbol{\Theta})\, P(\mathbf{y} \mid \boldsymbol{\Theta})\right] = \ln \prod_{n=1}^{N} p\,(\mathbf{x}_n \mid \mathbf{y},\ \boldsymbol{\Theta})\ P\,(\mathbf{y}_n \mid \boldsymbol{\Theta}) \\ &= \ln \prod_{n=1}^{N} \pi_{y_n}\, p\,(\mathbf{x}_n \mid \boldsymbol{\theta}_{y_n}) = \sum_{n=1}^{n} \ln\left[\pi_{y_n}\, p\,(\mathbf{x}_n \mid \boldsymbol{\theta}_{y_n})\right]\ . \end{aligned}$$

L'algorithme EM [Dempster77] effectue la mise à jour des paramètres $\boldsymbol{\Theta}$ par l'itération des deux étapes suivantes :

1. (*Expectation*) calculer la moyenne statistique conditionnée suivante :

$$E\left[\ln p(\mathbf{x},\ \mathbf{y} \mid \boldsymbol{\Theta}) \mid \mathbf{x},\ \widehat{\boldsymbol{\Theta}}^{(i)}\right] \overset{\text{not}}{=} Q(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)})\ , \tag{4.10}$$

   où $\widehat{\boldsymbol{\Theta}}^{(i)}$ représente les paramètres estimés à l'itération précédente ;

2. (*Maximisation*) maximiser $Q(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)})$ par rapport à $\boldsymbol{\Theta}$ et assigner la valeur obtenue au nouvel estimé :

$$\widehat{\boldsymbol{\Theta}}^{(i+1)} = \arg\max_{\boldsymbol{\Theta}} Q(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)})\ .$$

La succession de ces deux étapes garantit la croissance de la log-vraisemblance (4.9) entre deux itérations successives ainsi que la convergence linéaire vers un maximum local de cette fonction [Wu83].

[4] Dans la suite P désigne la probabilité.

Présentons maintenant les étapes de l'algorithme EM pour l'estimation des paramètres de la distribution de mélange. L'étape *Expectation* consiste à expliciter la fonction Q. Observons tout d'abord que si Q est traitée comme une fonction des données manquantes $\mathbf{y}$, alors la moyenne dans 4.10 s'exprime par :

$$E\left[\ln p(\mathbf{x},\, \mathbf{y} \mid \boldsymbol{\Theta}) \mid \mathbf{x},\, \widehat{\boldsymbol{\Theta}}^{(i)}\right] = \sum_{y \in \mathcal{Y}} P(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\Theta}}^{(i)}) \underbrace{\ln p(\mathbf{x},\, \mathbf{y} \mid \boldsymbol{\Theta})}_{L_{\mathrm{c}}(\boldsymbol{\Theta})} ,$$

où $\mathcal{Y}$ représente l'ensemble des valeurs de $\mathbf{Y}$. $L_{\mathrm{c}}(\boldsymbol{\Theta})$ étant explicité, il nous reste à exprimer la probabilité conditionnelle $P(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\Theta}}^{(i-1)})$. Utilisant la formule de Bayes, celle-ci s'écrit :

$$\begin{aligned} P\left(\mathbf{y} \mid \mathbf{x}, \widehat{\boldsymbol{\Theta}}^{(i)}\right) &= \frac{p(\mathbf{x} \mid \mathbf{y}, \widehat{\boldsymbol{\Theta}}^{(i)})\, P(\mathbf{y} \mid \widehat{\boldsymbol{\Theta}}^{(i)})}{p(\mathbf{x} \mid \widehat{\boldsymbol{\Theta}}^{(i)})} = \frac{\prod_{n=1}^{N} p(\mathbf{x}_n \mid \mathbf{y}, \widehat{\boldsymbol{\Theta}}^{(i)})\, P(y_n \mid \widehat{\boldsymbol{\Theta}}^{(i)})}{\sum_{k=1}^{K} \pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_k^{(i)}\right)} \\ &= \prod_{n=1}^{N} \frac{\pi_{y_n}^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_{y_n}^{(i)}\right)}{\sum_{k=1}^{K} \pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_k^{(i)}\right)} . \end{aligned}$$

Les quantités

$$\frac{\pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_k^{(i)}\right)}{\sum_{k=1}^{K} \pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_k^{(i)}\right)} \overset{\text{not}}{=} t_{nk}^{(i+1)} \tag{4.11}$$

sont les probabilités d'appartenance *a posteriori* des observations aux classes connaissant les paramètres de l'itération i.

L'étape *Maximisation* de l'algorithme EM consiste à maximiser

$$Q(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)}) = \sum_{y \in \mathcal{Y}} \left\{ \prod_{n=1}^{N} \frac{\pi_{y_n}^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_{y_n}^{(i)}\right)}{\sum_{k=1}^{K} \pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_k^{(i)}\right)} \sum_{n=1}^{n} \ln\left[\pi_{y_n}\, p\left(\mathbf{x}_n \mid \boldsymbol{\theta}_{y_n}\right)\right] \right\}$$

sous la contrainte $\sum_{k=1}^{K} \pi_k = 1$. *i.e.* à résoudre le système qui en résulte par l'annulation des dérivées partielles du Lagrangien :

$$l(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)}) = Q(\boldsymbol{\Theta}, \widehat{\boldsymbol{\Theta}}^{(i)}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

et à vérifier que la matrice hessienne de L soit définie négative. Cela est faisable pour certaines types de distribution de mélange. Typiquement, pour le mélange gaussien la résolution analytique conduit à :

$$\pi_k^{(i+1)} = \frac{1}{N} \sum_{n=1}^{N} t_{nk}^{(i+1)} \ , \tag{4.12}$$

$$\boldsymbol{\mu}_k^{(i+1)} = \frac{\sum_{n=1}^{N} t_{nk}^{(i+1)} \, \mathbf{x}_n}{\sum_{n=1}^{N} t_{nk}^{(i+1)}} \ , \tag{4.13}$$

$$\boldsymbol{\Sigma}_k^{(i+1)} = \frac{\sum_{n=1}^{N} t_{nk}^{(i+1)} \, (\mathbf{x}_n - \boldsymbol{\mu}_k^{(i+1)})^t \cdot (\mathbf{x}_n - \boldsymbol{\mu}_k^{(i+1)})}{\sum_{n=1}^{N} t_{nk}^{(i+1)}} \ . \tag{4.14}$$

En résumé, l'algorithme EM pour l'estimation des paramètres d'une distribution de type mélange gaussien est le suivant :

1. $i = 0$; initialiser les proportions $\pi_k^{(0)}$, les moyennes $\boldsymbol{\mu}_k^{(0)}$ et les matrices de covariance $\boldsymbol{\Sigma}_k^{(0)}$, $k = 1, \ldots, K$;
2. (étape *Expectation*) pour tout $n = 1, \ldots, N$ et $k = 1, \ldots, K$ calculer les probabilités d'appartenance *a posteriori* avec la formule (4.11) ;
3. (étape *Maximisation*) pour tout $k = 1, \ldots, K$ mettre à jour les proportions avec la formule (4.12), les moyennes avec la formule (4.13) et les matrices de covariance avec la formule (4.14) ;
4. *si* (CRITERE ARRET)

   retourner $\widehat{\boldsymbol{\Theta}}^{(i+1)} = \left(\left(\boldsymbol{\mu}_1^{(i+1)}, \boldsymbol{\Sigma}_1^{(i+1)} \right), \ldots, \left(\boldsymbol{\mu}_K^{(i+1)}, \boldsymbol{\Sigma}_K^{(i+1)} \right), \pi_1^{(i+1)}, \ldots, \pi_K^{(i+1)} \right)$;

   ARRET ;

   sinon

   aller à 2 ;

L'étape d'initialisation peut s'effectuer par une des procédures suivantes :

- donner des valeurs arbitraires aux paramètres $\pi_k^{(0)}$, $\boldsymbol{\mu}_k^{(0)}$ et $\boldsymbol{\Sigma}_k^{(0)}$ ou
- initialiser une partition dure $(y_1^{(0)}, \ldots, y_N^{(0)})$ et calculer les paramètres $\pi_k^{(0)}$, $\boldsymbol{\mu}_k^{(0)}$ et $\boldsymbol{\Sigma}_k^{(0)}$ en fonction de celle-ci.

Le critère d'arrêt peut être lié à la variation soit des paramètres estimés, soit des probabilités d'appartenance *a posteriori* entre deux cycles consécutifs.

En général, le résultat fourni par l'algorithme, correspondant à la détection d'un maximum local de la log-vraisemblance dépend de l'étape d'initialisation. Dans la section 4.4.1 nous présenterons une stratégie d'initialisation robuste pour la classification automatique des régions du visage dans des images couleurs.

4.2.3.2.c Analogie avec la classification floue. Régularisation spatiale Hathaway [Hathaway86] a démontré que l'algorithme EM appliqué à un modèle de mélange effectue les mêmes calculs qu'un classificateur flou dont la fonctionnelle-objectif est donnée par :

$$F(\mathbf{C}, \mathbf{\Theta}) = \sum_{k=1}^{K} \sum_{n=1}^{N} c_{ik} \, \ln(\pi_k \, p(\mathbf{x}_n \mid \boldsymbol{\theta}_k)) - \sum_{k=1}^{K} \sum_{n=1}^{N} c_{nk} \, \ln(c_{nk}) \; . \tag{4.15}$$

où $\mathbf{C} = (c_{nk})_{\substack{1 \leq n \leq N \\ 1 \leq k \leq K}}$, avec $0 \leq c_{nk} \leq 1$ et $\sum_{k=1}^{K} c_{nk} = 1$ pour tout $n = 1, \ldots, N$, représente une matrice de classification floue, à estimer simultanément avec les proportions π_k et les parametres $\boldsymbol{\theta}_k$. En effet, en annulant les dérivées du Lagrangien de $F(\mathbf{C}, \mathbf{\Theta})$, obtenu à partir des contraintes $\sum_{k=1}^{K} c_{nk} = 1$, on obtient le même résultat que dans l'étape *Expectation* de l'algorithme EM, notamment les éléments c_{nk} de la matrice de classification sont données par les mêmes formules que les probabilités t_{nk}.

A partir de ce formalisme, Ambroise [Ambroise97] suggère d'introduire un terme de contrainte spatiale dans (4.15), afin de prendre en compte les relations de voisinage spatial entre les observations appartenant à la même classe. Ainsi, la nouvelle fonctionnelle-objectif est exprimée par :

$$G(\mathbf{C}, \mathbf{\Theta}) = F(\mathbf{C}, \mathbf{\Theta}) + \frac{1}{2} \beta \sum_{k=1}^{K} \sum_{m=1}^{N} \sum_{n=1}^{N} c_{mk} \; c_{nk} \, v_{mn} \; , \tag{4.16}$$

avec

$$v_{mn} = \begin{cases} 1, & \text{si } \mathbf{x}_m \text{ et } \mathbf{x}_n \text{ sont voisins} \\ 0, & \text{sinon} \end{cases}$$

et β un paramètre empirique de régularisation. Intuitivement ce nouveau terme sera d'autant plus grand que les classes contiendront d'avantage d'éléments avoisinants. Ainsi, les configurations spatialement régulières seront privilégiées. Imposant les conditions nécessaires de maximum local pour 4.16 dans le cas d'un mélange gaussien, on obtient les formules suivantes de

mise à jour des éléments de la matrice de classification (l'étape *Expectation*).

$$c_{nk}^{(i+1)} = \frac{\pi_k^{(i)}\, p\left(\mathbf{x}_n \mid \left(\boldsymbol{\mu}_k^{(i)},\, \boldsymbol{\Sigma}_k^{(i)}\right)\right) \exp\left(\beta \sum\limits_{m=1}^{N} c_{mk}^{(i+1)}\, v_{nm}\right)}{\sum\limits_{l=1}^{K} \left[\pi_l^{(i)}\, p\left(\mathbf{x}_n \mid \left(\boldsymbol{\mu}_l^{(i)},\, \boldsymbol{\Sigma}_l^{(i)}\right)\right) \exp\left(\beta \sum\limits_{m=1}^{N} c_{ml}^{(i+1)}\, v_{nm}\right)\right]} \;,\; n = 1, \ldots, N,\; k = 1, \ldots, K \;. \tag{4.17}$$

En outre, l'étape *Maximisation* reste identique à celle de l'algorithme EM. Remarquons que 4.17 donne le calcul implicite de la matrice de classification, sous la forme :

$$\mathbf{C}^{(i+1)} = \mathbf{g}(\mathbf{C}^{(i+1)}) \;, \tag{4.18}$$

ce qui suggère le calcul de cette matrice en itérant $\mathbf{g}$, dans le cas où cette dernière possède un point fixe. En effet, l'algorithme est convergent lorsque le paramètre de régularisation β ne dépasse pas un certain seuil [Ambroise98]. Par ailleurs, pour $\beta = 0$, on obtient l'algorithme EM, puisque $G(\mathbf{C}, \boldsymbol{\Theta}) = F(\mathbf{C}, \boldsymbol{\Theta})$.

Cet algorithme, nommé Neighborhood EM (abrégé NEM), présente quelques avantages : il ne pose pas de problèmes de réglage de paramètres et est comparable en termes de temps de calcul à l'algorithme EM, puisque (4.18) présente une convergence très rapide (typiquement 1-2 itérations).

4.3 Prototypes déformables compatibles MPEG-4

La demande croissante de diffusion de contenus visuels de haute qualité sur les réseaux existants a conduit au développement de nouveaux schémas de codage et compression capables de réaliser le bon compromis entre bande de transmission limitée et qualité visuelle du contenu.

Dans ce contexte, le standard MPEG-4 propose une approche orientée objet de codage et transmission de scènes vidéos. Une scène est ainsi découpée en plusieurs objets (arrière-plan ou décor et les différents personnages qui y évoluent) dont les formes et les paramètres de mouvement sont codés et transmis pour être synthétisés au niveau du décodeur. Il suffit ainsi de transmettre une seule fois les paramètres de chaque objet pour recréer ensuite la scène à l'aide du flux de données contenant les paramètres de mouvement.

Cette technique nécessite de disposer d'une description paramétrique des objets codés, sous forme de modèles enrichis par une information de déformation. Dans le contexte d'animation d'avatars, la norme MPEG-4 utilise une spécification de codage pour l'animation d'humanoïdes virtuels - *Face and Body Animation* (FBA), dont nous allons détailler par la suite le descripteur de visage.

4.3.1 Descripteur de visage MPEG-4

Conceptuellement, le descripteur de visage MPEG-4 réunit deux classes de paramètres : les paramètres de description du modèles de visage en termes de forme et de texture, *Face Definition Parameters* (FDPs), et les paramètres d'animation du modèle, *Face Animation Parameters* (FAPs), qui précisent le movement global et/ou les déformations locales du visage. Le descripteur donne ainsi la possibilité de reproduire un large éventail d'expressions faciales, d'émotions ou les mouvements de la parole.

4.3.1.1 Paramètres de description du visage (FDPs)

Pour assurer un rendu réel des expressions faciales lors de l'étape d'animation, le descripteur de visage définit un jeu de paramètres (FDPs) caractérisant d'une part, la forme du modèle de visage à l'aide d'un maillage 3D et, d'autre part, la texture associée (Figure 4.2). A cette description s'ajoute un ensemble de points caractéristiques qui seront directement affectés par les paramètres d'animation (FAPs), leur emplacement sur le visage (Figure 4.3) leur permettant de simuler une large variété d'expressions. L'avantage d'introduire ces points réside dans la possibilité d'utiliser des modèles différents au niveau du codeur et du décodeur pour jouer la même scène. Toutefois, lorsque la totalité des paramètres de modèle est transmis dans une phase d'initialisation, cela permet d'individualiser le modèle existant au niveau du décodeur et d'accroître la précision du rendu de certaines expressions faciales.

4.3.1.2 Paramètres d'animation du visage (FAPs)

Les paramètres d'animation du visage (FAPs) spécifient l'amplitude et la direction du déplacement des points caractéristiques de FDPs lors de l'animation. Ils sont issus d'une étude sur les actions faciales minimales et sont étroitement liés aux actions musculaires. Ils représentent un ensemble complet d'actions faciales de base et, par conséquent, permettent le rendu de la majorité des expressions naturelles.

Les FAPs se divisent en deux classes de paramètres de haut niveau : les *visèmes* et les *expressions*. Un visème représente l'équivalent visuel d'un phonème, notamment l'ensemble des déformations du visage correspondant à la prononciation d'un phonème. Actuellement, la norme MPEG-4 inclut seulement les visèmes statiques qui sont clairement distingués, mais d'autres pourront être ajoutés dans une extension future du standard. Le Tableau 4.1 présente les phonèmes ayant des visèmes définis dans le standard.

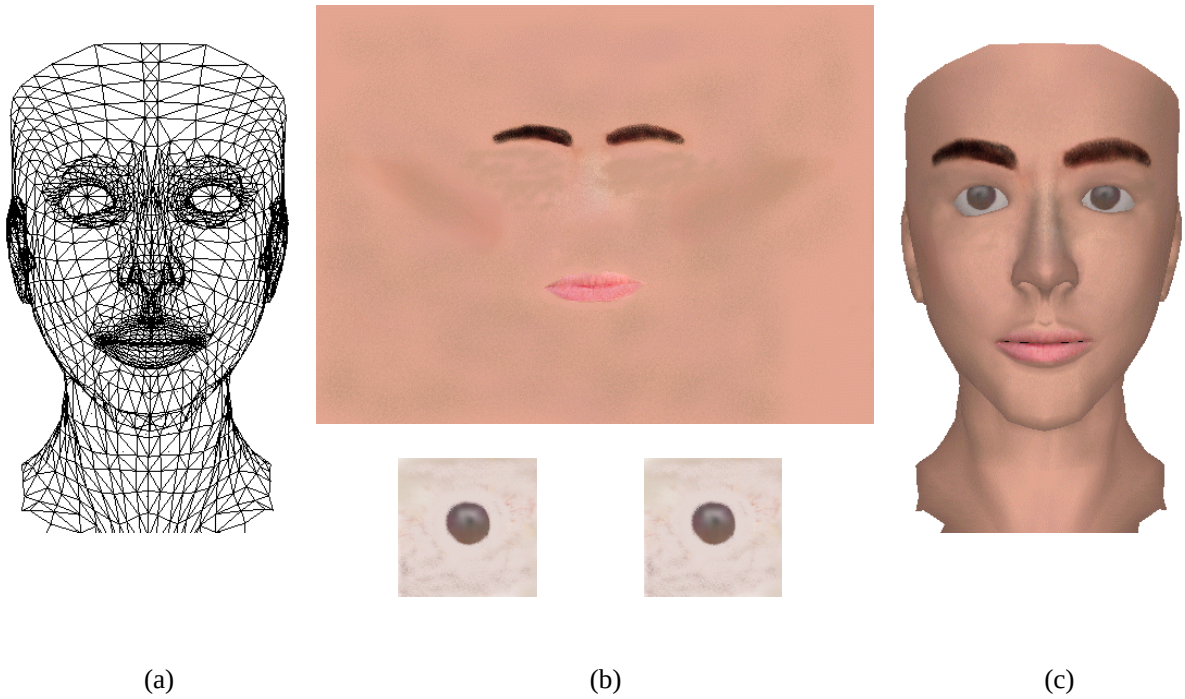


Figure 4.2 : Les paramètres FDPs regroupant (a) la description de forme par maillage 3D et (b) l'information de texture (ici la peau et les yeux) pour générer le modèle de visage texturé (c).

Numéro du visème	phonème	exemple
1	p, b, m	put, bed, mill
2	f, v	far, voice
3	T, D	think, that
4	t, d	tip, doll
5	k, g	call, gas
6	tS, dZ, S	chair, join, she
7	s, z	sir, zeal
8	n, l	not, lot
9	r	red
10	A:	car
11	e	bed
12	I	tip
13	O	top
14	U	book

Tableau 4.1 : Les visèmes définis dans la norme MPEG-4.

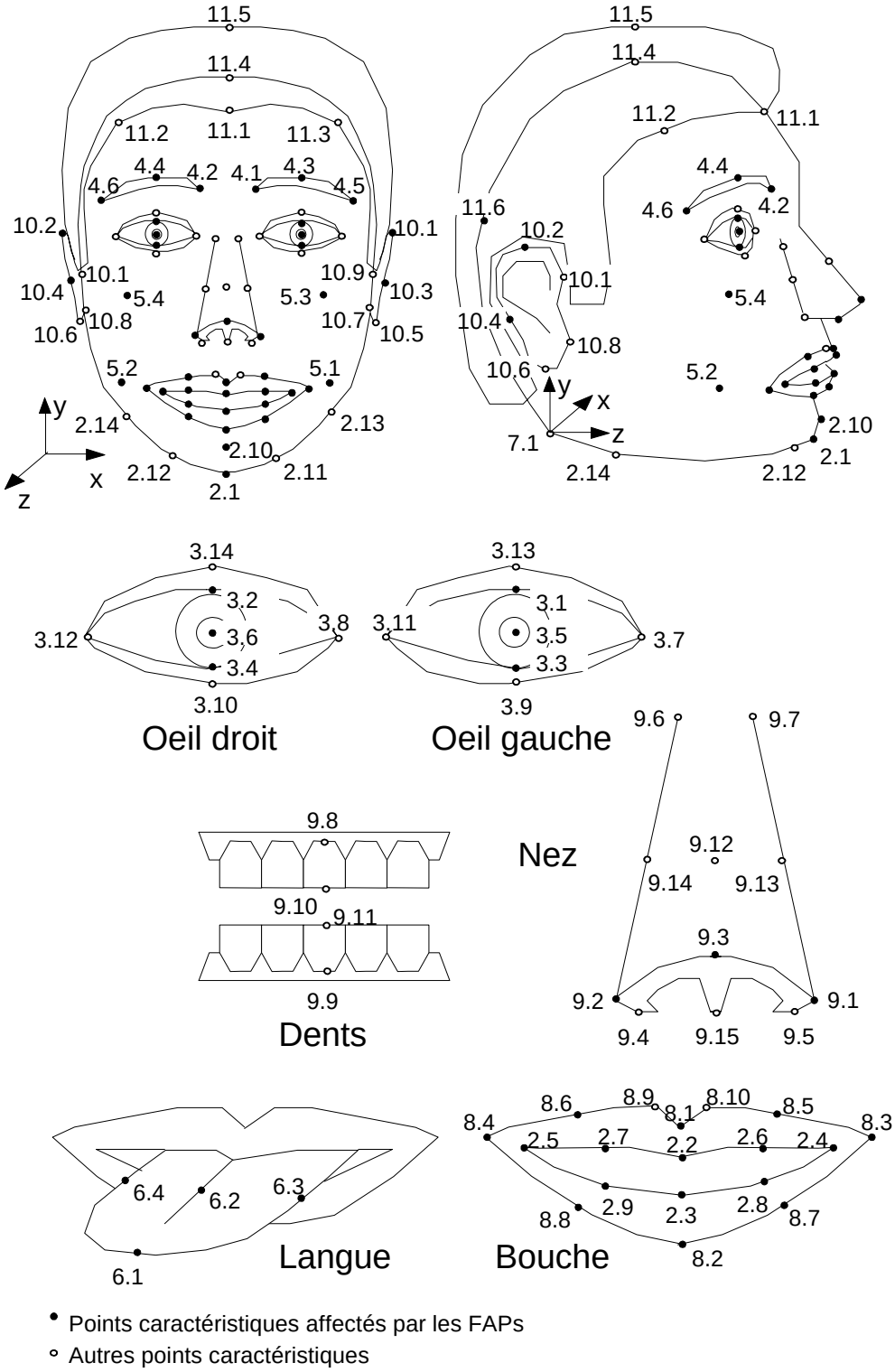


Figure 4.3 : Points caractéristiques des FDPs tels que définis dans la norme MPEG-4.

De manière similaire, les paramètres d'expression permettent la définition d'expressions faciales de haut niveau. Les valeurs que ces paramètres peuvent prendre sont spécifiées par description textuelle, comme le montre le Tableau 4.2 ci-dessous.

Expression	Description textuelle
joy	Les sourcils sont décontractés. La bouche est ouverte et les commissures des lèvres retirées en arrière, vers les oreilles.
sadness	Les coins intérieurs des sourcils sont courbés vers le haut. Les yeux sont légèrement fermés. La bouche est décontractée.
anger	Les coins intérieurs des sourcils sont abaissés ensemble. Les yeux sont largement ouverts. Les lèvres sont serrées l'une contre l'autre ou ouvertes pour montrer les dents.
fear	Les sourcils sont levés ensemble et leur partie intérieure est courbée vers le haut. Les yeux sont contractés et en état d'alerte.
disgust	Les sourcils et les paupières sont décontractés. La lèvre supérieure est levée et courbée, souvent de manière asymétrique.
surprise	Les sourcils sont levés. Les paupières supérieures sont ouvertes, les paupières inférieures, décontractées. La bouche est ouverte.

Tableau 4.2 : Les expressions faciales définies dans la norme MPEG-4 et leur description textuelle.

Pour faciliter l'animation du visage, les FAPs qui peuvent être utilisés ensemble pour représenter des expressions naturelles, sont rassemblés en groupes qui sont référencés à l'aide d'un paramètre d'expression.

Pour assurer l'interprétation uniforme des FAPs sur un modèle de visage compatible MPEG-4 quelconque, en termes de rendu cohérent d'expressions ou de parole, les FAPs impliquant un mouvement de translation sont exprimés en unités de mesure spécifiques, dénommées FAPU (*Facial Animation Parameter Units*), permettant de s'affranchir d'une étape de calibrage préalable du modèle.

Les FAPU sont définies comme des fractions des distances entre certains points caractéristiques de l'ensemble FDP et sont choisis pour assurer une précision suffisante des mouvements d'animation. Ces unités sont illustrées dans le Tableau 4.3, avec les notations des Figures 4.3 et 4.4.

FAPU	Caractéristique faciale de référence
IRISD = IRISD0 / 1024	IRISD0 = 3.1.y - 3.3.y
ES = ES0/1024	ES0 = 3.5.x - 3.6.x
ENS = ENS0/1024	ENS0 = 3.5.y - 9.15.y
MNS = MNS0/1024	MNS0 = 9.15.y - 2.2.y
MW = MW0/1024	MW0 = 8.3.x - 8.4.x
AU = 10^{-5} rad (unité angulaire)	

Tableau 4.3 : *Facial Animation Parameter Units* (FAPU) définies en fonction des distances entre des points caractéristiques du visage (Figure 4.4) dont l'identification est donnée Figure 4.3. Ici, 3.1.y représente la coordonnée y du point 3.1.

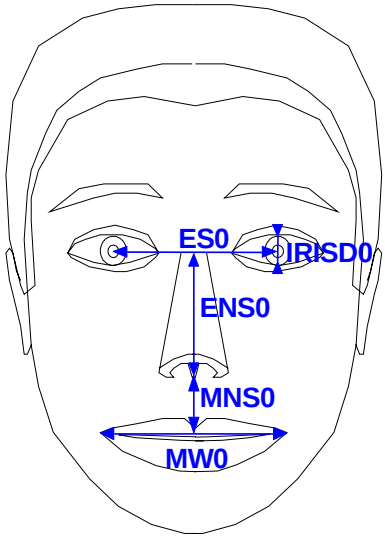


Figure 4.4 : Caractéristiques faciales de référence pour la définition des *Facial Animation Parameter Units* (FAPU).

Les FAPs sont exprimés sous forme de déplacements par rapport à une position de référence du visage, appelée visage neutre, définie comme suit :

- le système de coordonnées est droit, les axes étant parallèles aux axes du système fixe de référence ;
- le regard est dirigé dans la direction de l'axe z (Figure 4.3) ;
- tous les muscles du visage sont relâchés ;
- les paupières sont tangentes à l'iris ;
- le diamètre de la pupille est le tiers de IRISD0 (Figure 4.4) ;

- les lèvres sont en contact, la ligne de séparation horizontale est à la même hauteur que les coins des lèvres ;
- la bouche est fermée et les dents supérieures touchent les dents inférieures ;
- la langue est applatie, en position horizontale, avec le bout touchant la frontière entre les dents supérieures et inférieurs (le point caractéristique 6.1 touche le 9.11, Figure 4.3).

4.3.2 Construction de prototypes déformables compatibles MPEG-4

Comme nous l'avons précisé dans la section précédente, le standard MPEG-4 laisse aux utilisateurs le libre choix d'un modèle de visage dans le cadre d'applications d'animation faciale, en imposant uniquement la présence de l'ensemble de points caractéristiques permettant le contrôle de l'animation à travers le flux de FAPs.

Dans le cadre de notre application de suivi des mouvements locaux du visage, nous avons défini deux prototypes déformables compatibles MPEG-4, que nous présentons par la suite.

4.3.2.1 Modélisation géométrique

Dans l'approche proposée, le prototype adapté à la bouche noté $\mathcal{P}_{\text{bouche}}$ est défini comme suit :

$$\mathcal{P}_{\text{bouche}} = \mathcal{L}_{\text{ss}} \cup \mathcal{L}_{\text{si}} \cup \mathcal{L}_{\text{is}} \cup \mathcal{L}_{\text{ii}} \ ,$$

où $\mathcal{L}_{\text{ss}}$ (respectivement $\mathcal{L}_{\text{si}}$) désigne le contour supérieur (respectivement inférieur) de la lèvre supérieure et $\mathcal{L}_{\text{is}}$ (respectivement $\mathcal{L}_{\text{ii}}$) le contour supérieur (respectivement inférieur) de la lèvre inférieure (Figure 4.5 ci-dessous).

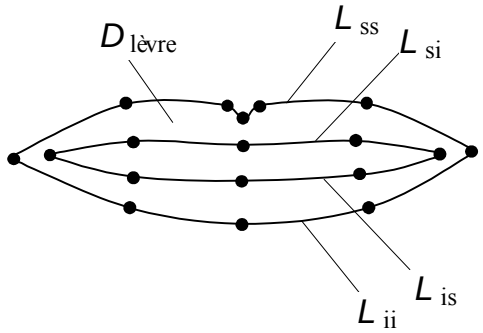


Figure 4.5 : Modèle géométrique du prototype de la bouche.

$\mathcal{L}_{ab}$avec $a, b \in \{s, i\}$ représentent des courbes 2D interpolant les points caractéristiques MPEG-4 associés à la bouche (les points 2.2÷2.9 et 8.1÷8.10, Figure 4.3). Notons par $\mathcal{D}_{\text{lèvre}}$ le domaine correspondant à la réunion des deux lèvres.

Dans le cas de l'œil, il convient de prendre en compte la configuration ouverte ou fermée de l'œil en raison de la rupture photométrique qu'elle introduit. C'est pourquoi nous définissons pour chacune de ces configurations un prototype spécifique :

$$\mathcal{P}_{\text{il ouvert}} = \mathcal{P}_{\text{s}} \cup \mathcal{P}_{\text{i}} \cup \mathcal{I} \ ,$$

$$\mathcal{P}_{\text{il fermé}} = \mathcal{P}_{\text{s}} \cup \mathcal{P}_{\text{i}} \ ,$$

où $\mathcal{P}_{\text{s}}$ (respectivement $\mathcal{P}_{\text{i}}$) désigne le contour de la paupière supérieure (respectivement inférieure) interpolant les points caractéristiques MPEG-4 correspondants et $\mathcal{I}$ représente le contour de l'iris, (Figure 4.6). Les deux régions distinctes de l'œil ouvert, le blanc et l'iris sont désignées respectivement par $\mathcal{D}_{\text{blans}}$ et $\mathcal{D}_{\text{iris}}$. Précisons que notre modèle suppose une épaisseur nulle des paupières, impliquant l'identité des points suivants (Figure 4.3) : $3.1 \equiv 3.13$, $3.3 \equiv 3.9$, $3.2 \equiv 3.14$, $3.4 \equiv 3.10$.

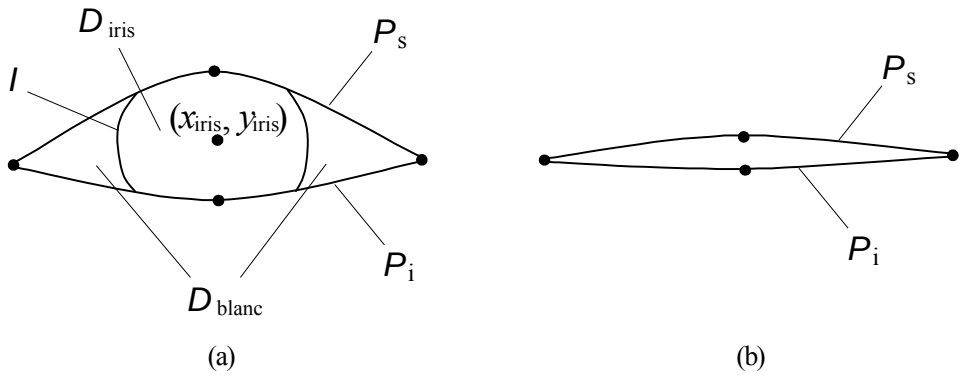


Figure 4.6 : Modèles géométriques des prototypes de l'œil ouvert (a) et de l'œil fermé (b).

Le choix de l'interpolateur qui génère les courbes ci-dessus doit assurer un bon compromis entre précision et compacité de la représentation. Les polynômes du deuxième degré et notamment les arcs de parabole, en constituent une possibilité [Reinders95, Yuille92, Zhang.L98]. Toutefois, cette représentation quadratique se révèle par trop rigide pour gérer la grande richesse des expressions faciales, notamment pour des déformations accentuées des lèvres. En effet, pour des déformations symétriques et d'amplitude raisonnable, ces modèles, paramétrés par trois points de contrôle, se comportent bien (Figure 4.7), mais une fois franchie la limite pour laquelle l'approximation parabolique est acceptable, cette représentation n'assure plus une modélisation fiable (Figure 4.8).

Un autre type d'approche s'appuie sur une modélisation harmonique des contours [Leroy95]. Même si celle-ci offre une certaine souplesse, la représentation harmonique ne permet pas une

Figure 4.7 : Succès de la modélisation des contours des lèvres par des arcs de paraboles : (a), (b) points de contrôle, (c), (d) ajustement.

description géométrique intuitive et immédiate au niveau des déformations.

Toutefois, à notre avis, la représentation à base de splines reste dans ce contexte le meilleur choix. Nous avons donc adopté cette approche et modélisé les contours précédemment définis par des B-splines cubiques interpolant les points caractéristiques considérés. Dans le cas particulier de l'œil ouvert, le contour de l'iris est représenté par un cercle de centre $(x_{\text{iris}}, y_{\text{iris}})$ coupé par les splines correspondant aux paupières (Figure 4.6 (a)).

Détaillons la procédure d'interpolation sur la configuration présentée Figure 4.9. $\mathbf{p}_i = (x_i, y_i)$, $i = 1, \ldots, n$ désignent les points à interpoler et $l_i = \|\mathbf{p}_{i+1} - \mathbf{p}_i\|$, $i = 1, \ldots, n-1$ représentent les longueurs des segments reliant des points consécutifs.

Définissons les coordonnées curvilignes des points sur la ligne brisée $\mathbf{p}_1\mathbf{p}_2 \ldots \mathbf{p}_n$ par :

$$u_1 = 0 ,$$

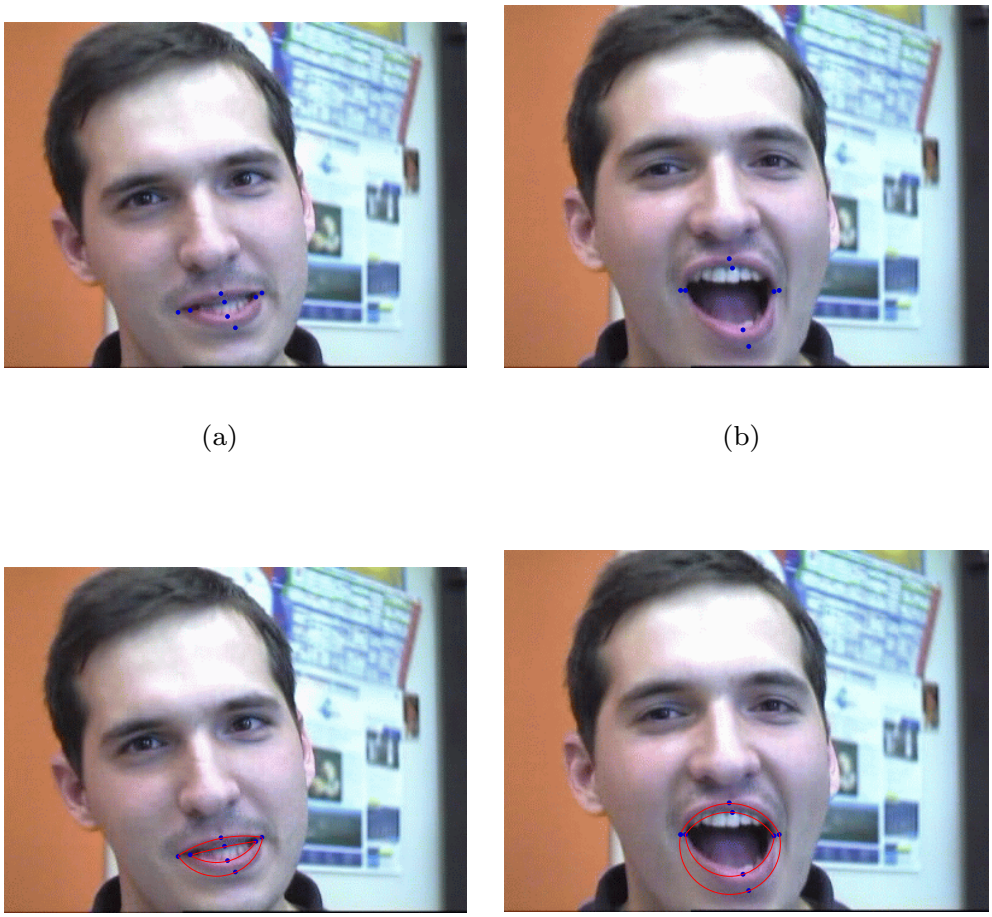


Figure 4.8 : Echec de la modélisation des contours des lèvres par des arcs de paraboles : (a), (b) points de contrôle, (c), (d) ajustement.

$$u_i \quad = \quad \sum_{j=1}^{i-1} l_j \,/\, \sum_{j=1}^{n-1} l_j \in (0,\, 1] \ , \ i = 2,\, \ldots,\, n \ .$$

Ainsi, nous obtenons deux ensembles de couples, $\{(u_i,\, x_i) \mid i = 1,\, \ldots,\, n\}$ $\{(u_i,\, y_i) \mid i = 1,\, \ldots,\, n\}$. Appliquons à chacun la technique d'interpolation à base de B-splines décrite dans la section 4.2.2.2. Selon le théorème 4.2, pour interpoler n points avec des B-splines cubiques, il nous faut $n+4$ nœuds. Le même théorème assure l'existence et l'unicité de l'interpolateur dans le cas où le nombre de nœuds consécutifs identiques est inférieur ou égal à 3. Ainsi, les 3 premiers et les 3 derniers nœuds (les nœuds externes) sont choisis identiques à u_0 et u_n, respectivement :

$$\begin{aligned} t_1 \quad &= \quad t_2 = t_3 = 0 \ , \\ t_{n+2} \quad &= \quad t_{n+3} = t_{n+4} = 1 \ . \end{aligned}$$

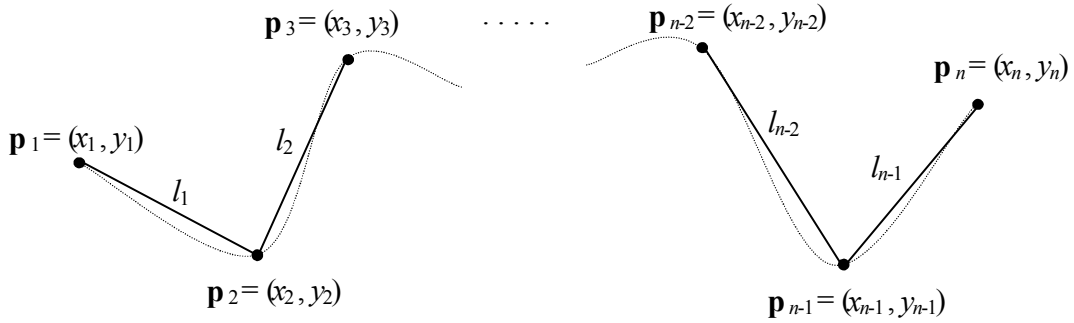


Figure 4.9 : Interpolation d'un ensemble de points du plan par une courbe spline.

En conséquence, il nous reste $n-2$ nœuds à distribuer dans l'intervalle $[0,1]$. Pour cela nous utilisons la méthode de De Boor [DeBoor78], définissant les nœuds internes comme la moyenne des coordonnées curvilignes sur un support de longueur égale au degré de la spline interpolante. Dans le cas de la spline cubique, ces nœuds sont donnés par :

$$t_{i+3} = \frac{1}{3} \sum_{j=i}^{i+2} u_j \ , \ i = 1, \ldots, n-2 \ .$$

Le calcul des interpolateurs $x(t)$ et $y(t)$ consiste à déterminer tout d'abord les B-splines associées aux nœuds ainsi définis (les formules (4.3)) et à résoudre ensuite les deux systèmes linéaires, de l'équation (4.4) appliquée respectivement pour $x(u_i) = x_i$ et $y(u_i) = y_i$, $1, \ldots, n$.

Ce type de modélisation permet d'intégrer tous les points caractéristiques spécifiés par le standard MPEG-4 dans le cadre d'une représentation souple et précise, comme le montrent les images présentées Figure 4.10, correspondant à diverses expressions faciales normales ou exagérées.

4.3.2.2 Contraintes internes

Afin de maintenir la variabilité de la forme du prototype dans des limites imposées par les déformations de l'objet modélisé, des contraintes internes sont introduites sous forme d'une fonctionnelle d'énergie comportant une composante élastique et une composante liée aux contraintes de symétrie locale.

Pour spécifier le terme d'élasticité, considérons les deux graphes présentés Figure 4.11, dont les sommets sont respectivement les points caractéristiques de la bouche et de l'œil ouvert. Les arêtes décrivent l'interdépendance des déformations au niveau des points caractéristiques. En remplaçant les arêtes par des ressorts, on obtient des systèmes physiques de déformation permettant de formaliser l'énergie interne des prototypes.

Figure 4.10 : Modélisation de la bouche et des yeux par des B-splines cubiques reliant les points caractéristiques MPEG-4 (en bleu), pour différentes expressions faciales plus ou moins exagérées.

Soit le ressort j, au repos de longueur l_j, et après mise sous tension par une transformation τ, de longueur l'_j. Dans ce cas, l'énergie élastique du prototype déformé s'exprime, pour l'ensemble des ressorts, par :

$$E_{\text{élastique}} = \sum_j k_j \ (l_j - l'_j)^2 \ ,$$

où k_j désigne la constante d'élasticité du ressort j. Ici, les ressorts longitudinaux (le long des contours) et transversaux (inter-contours) respectivement sont supposés avoir la même constante d'élasticité, k_l (resp. k_t). De plus, on suppose $k_t \approx 10\, k_l$.

La composante énergétique exprimant certaines relations de symétrie vise à pénaliser la distribution non uniforme des points caractéristiques. Dans le cas du prototype de la bouche, chaque point caractéristique situé au milieu d'un contour de lèvre sera ‹‹encouragé››à rester

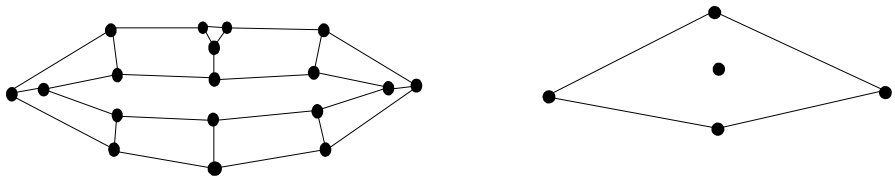


Figure 4.11 : Graphes associés aux points caractéristiques de la bouche et de l'œil ouvert, spécifiant l'interdépendance des déformations.

équidistant des commissures. De même, chaque point situé entre une commissure et le point du milieu de la lèvre sera ‹‹encouragé››à rester équidistant de ceux-ci. En utilisant les notations de la Figure 4.12, on définit l'énergie de symétrie suivante :

$$E_{\text{symétrie,bouche}} = \sum_{a,b\in\{\text{s,i}\}} \left[\left(\frac{l_{ab,\,0}}{l_{ab,\,1}} - 1 \right)^2 + \left(\frac{l_{ab,\,2}}{l_{ab,\,3}} - 1 \right)^2 + \left(\frac{l_{ab,\,4}}{l_{ab,\,5}} - 1 \right)^2 \right] .$$

Dans le cas de l'œil ouvert, le même principe de symétrie est appliqué, mais portant sur la moitié de la paupière et conduisant à l'expression suivante:

$$E_{\text{symétrie,œil ouvert}} = \sum_{a\in\{\text{s,i}\}} \left(\frac{l_{a,\,0}}{l_{a,\,1}} - 1 \right)^2 .$$

L'énergie interne du prototype s'exprime comme une somme pondérée des deux termes

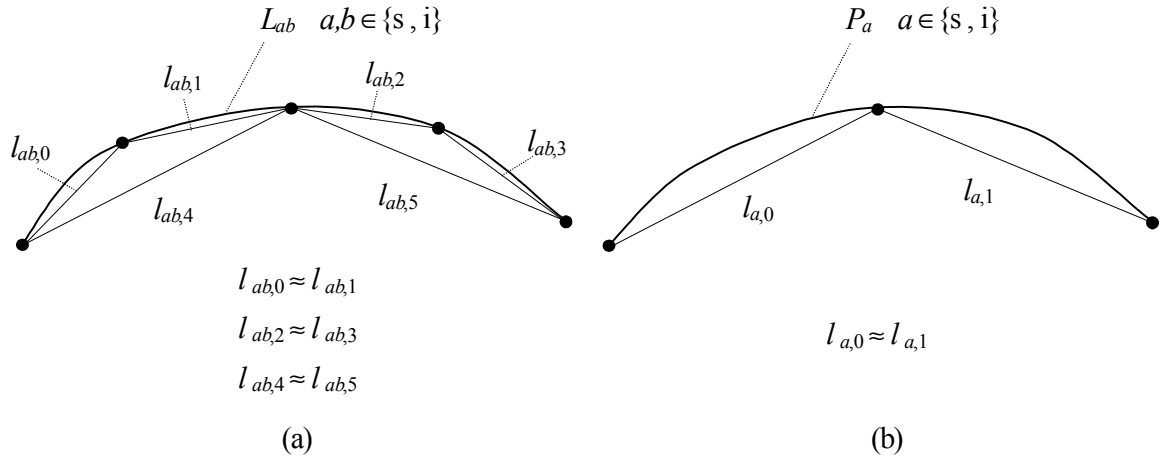


Figure 4.12 : Symétries locales exprimées sur un contour de lèvre (a) et d'œil (b).

ci-dessus définis :

$$E_{\text{int}} = c_{\text{élastique}}\, E_{\text{élastique}} + c_{\text{symétrie}}\, E_{\text{symétrie}} .$$

Les prototypes pour la bouche et l'œil ouvert ou fermé et leurs caractéristiques physiques

intrinsèques étant spécifiés, il convient maintenant de disposer d'une méthode capable de reconnaître si l'œil est en configuration ouverte ou fermée.

4.3.2.3 Détection de la configuration de l'œil

Pour cela, nous avons développé un algorithme automatique fondé sur la détection de la présence de l'iris dans l'image.

Si l'œil est ouvert, l'iris apparaît sous forme d'une petite région foncée, quasi-ronde, sur un fond clair (blanc de l'œil ou teinte chair). D'autre part, si l'œil est fermé, la petite région foncée correspondant aux cils sera de forme allongée. Détecter cette région revient à extraire les vallées locales dans une région d'intérêt associée à l'œil. Les approches utilisant les opérateurs morphologiques classiques de type black top hat, H-vallées ou RH-maxima [Serra94] nécessitent l'ajustement *ad-hoc* de paramètres en fonction des conditions d'éclairement, ce qui rend la mise au point d'une procédure automatique robuste assez difficile. En recourant au classificateur NEM pour un mélange gaussien, appliqué dans une région d'intérêt associée à l'œil, on s'affranchit de ces contraintes paramétriques tout en disposant d'une description quantitative en terme de coefficient d'appartenance à la classe "foncée".

Effectuer la classification dans un espace couleur rend le résultat plus robuste que dans le cas où seul le niveau de gris est pris en compte. Mentionnons que le résultat de la classification varie très peu par rapport aux différents espaces couleur considérés (RGB, Yuv, Lab, HSV). L'initialisation de l'algorithme NEM est effectuée par l'intermédiaire d'une partition initiale dure, établie en fonction de la luminance des pixels. De plus, le reflet cornéen est préalablement atténué à l'aide d'une ouverture morphologique marginale appliquée sur les trois composantes couleur.

La Figure 4.13 montre les résultats de la classification appliquée sur des images réelles d'œil ouvert ou fermé, obtenus pour différentes valeurs du paramètre de régularisation spatiale β. Dans les deux cas, le nombre *a priori* de classes est établi à 3. Remarquons la stabilité du résultat par rapport à la variation du paramètre β dès que la régularisation spatiale est imposée ($\beta > 0$).

L'ellipse d'inertie de la fonction d'appartenance à la classe d'intérêt ("foncée") donne la segmentation de l'iris ou des cils, selon le cas (Figure 4.13 (e) et (j)) et permet de décider de la configuration ouverte ou fermée de l'œil en fonction du rapport de ses deux axes.

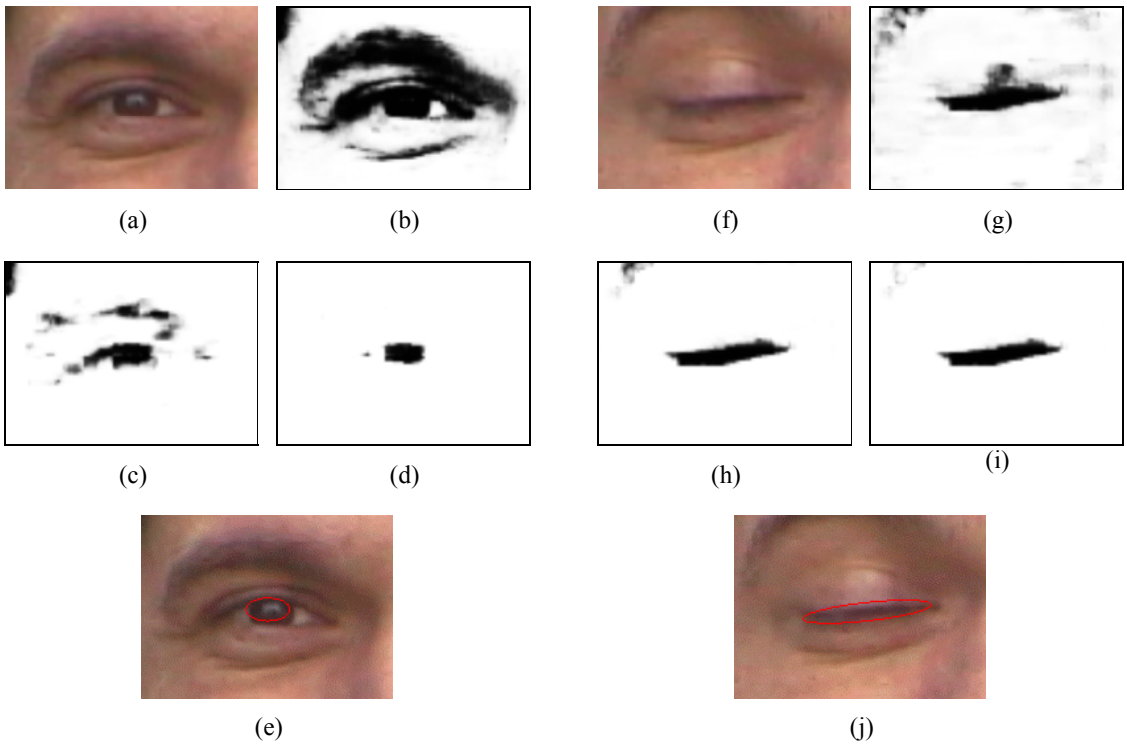


Figure 4.13 : Classification floue sous contrainte spatiale pour la détection de la configuration de l'œil. Images de test (a), (f), résultat de la classification˜: (b), (g) $\beta = 0$, (c), (h) $\beta = 0.1$, (d), (i) $\beta = 0.2$, et détection de la configuration ouverte/fermée de l'œil à l'aide de l'ellipse d'inertie (e), (j), ajustée respectivement sur (d), (i).

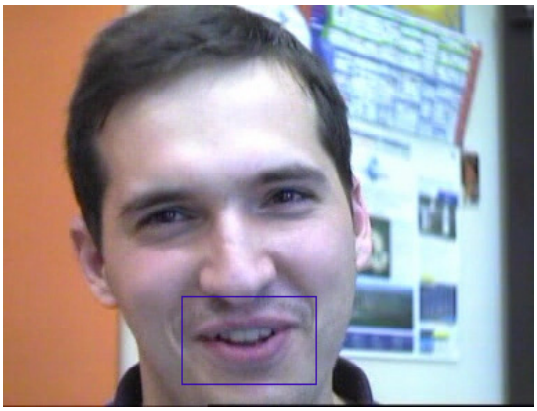
4.4 L'interaction prototype - image

Le recalage du prototype sur l'objet ciblé consiste en une adaptation de sa forme, décrite à l'aide d'une transformation géométrique 2D, notée τ. Cette adaptation s'effectue suite à une interaction avec l'image à travers une fonctionnelle d'énergie externe. Celle-ci établit une mesure d'appariement entre la géométrie du prototype et les primitives de l'image. Le point clef du succès du recalage à base de prototypes déformables consiste donc à définir et extraire des descripteurs pertinents et stables dans les régions d'intérêt de l'image, contenant la bouche et les yeux, et à les combiner au sein d'une fonctionnelle présentant un nombre réduit de minima locaux.

4.4.1 Définition et extraction des primitives d'image

Dans l'algorithme développé, la définition des primitives d'image envisageables pour spécifier l'interaction entre les prototypes et les données a été guidée par les observations suivantes.

La région des lèvres se caractérise par une chromaticité relativement uniforme, différente de celle correspondant à la peau ou à la partie inter-lèvres. Nous proposons d'utiliser à nouveau la procédure de classification automatique NEM pour un mélange gaussien, appliqué sur une région d'intérêt associée à la bouche. Cette classification est effectuée par rapport aux composantes couleur des pixels. Comme nous l'avons mentionné dans la section 4.2.3.2.b, le résultat de la classification dépend de l'étape d'initialisation. Une initialisation non adéquate risque de ne pas conduire à une segmentation précise de la région ciblée. En effet, la Figure 4.14 illustre la fonction d'appartenance à la classe "lèvres" estimée par l'algorithme NEM utilisant un nombre *a priori* de trois classes dont les paramètres sont initialisés de manière arbitraire.



(a)

(b)

Figure 4.14 : Classification obtenue par l'algorithme NEM appliqué pour trois classes dont les paramètres sont initialisés de manière arbitraire : (a) définition d'une région d'intérêt sur l'image de test ; (b) fonction d'appartenance à la classe "lèvres" (valeurs élevées en noir).

En pratique, il est difficile d'obtenir une initialisation robuste avec les seules informations fournies par l'image-même. Nous proposons donc de rajouter l'information liée à la pose 3D de la tête pour assigner les coefficients *a priori* d'appartenance à chaque classe, à partir du prototype recalé sur l'image précédente, comme suit :

- une région d'intérêt contenant le prototype recalé sur l'image précédente I_{n-1} est sélectionnée ; le prototype divise cette région en trois parties, correspondant aux lèvres, à la zone inter-lèvres et à la peau ;

- les composantes couleur des pixels situés dans ces trois parties sont utilisées pour appréhender les paramètres de trois distributions gaussiennes multivariées (les moyennes et les matrices de covariance) qui sont *a priori* associées à ces caractéristiques faciales ;
- utilisant le modèle 3D de tête et les paramètres de la pose 3D, la projection 2D/3D/2D de chacune des trois parties sur l'image courante I_n définit une partition dure sur celle-ci ;
- les paramètres des gaussiennes, d'une part, et la partition dure, d'autre part, donnent l'initialisation de l'algorithme NEM sur l'image I_n.

La procédure d'initialisation et le résultat de classification correspondant aux lèvres sont illustrés Figures 4.15-4.18 pour différentes déformations.

Les contours extérieurs des lèvres restent relativement stables par rapport aux déformations et aux changements des conditions d'éclairement. Le gradient de l'image peut donc être utilisé. En revanche, les contours intérieurs des lèvres ne peuvent pas être exploités directement puisque, en général, ils sont fortement influencés par les occultations présentes dans la région inter-lèvres. L'extraction du gradient de l'image est effectué par l'algorithme de Canny.

La texture des lèvres peut être prise en compte dans le cas où il n'y a pas de variations sévères d'éclairement ; sous cette hypothèse, la luminance de l'image peut être directement exploitée. Cette observation reste valable pour l'œil, mais sous la contrainte supplémentaire qu'il reste ouvert d'une image à l'autre. Les instabilités dues au reflet cornéen peuvent être éliminées par une ouverture morphologique en niveaux de gris.

L'information relative à la topographie locale de l'image, exprimée en termes de vallées et de dômes, peut être exploitée de manière efficace dans la région des yeux. Dans le cas où l'œil a été détecté en configuration ouverte sur l'image courante, le contraste relatif entre les zones correspondant à l'iris et au blanc de l'œil sera utilisé dans l'expression de l'énergie externe. Le cas contraire sera traité de manière à imposer au prototype une configuration prédéfinie, étiquetée "fermée".

4.4.2 Contraintes externes

A partir des remarques précédemment énumérées, les expressions des énergies externes associées à une séquence d'images $(I_n)_n$ sont définies en fonction de la transformation 2D qui adapte la forme du prototype (en utilisant les notations des Figures 4.5 et 4.6) comme suit :

$$\begin{aligned} E_{\text{ext,bouche}} &= c_{\text{texture,bouche}}\, E_{\text{texture,bouche}} + c_{\text{segmentation,bouche}}\, E_{\text{segmentation,bouche}} + \\ &\quad c_{\text{gradient,bouche}}\, E_{\text{gradient,bouche}} \ , \end{aligned}$$

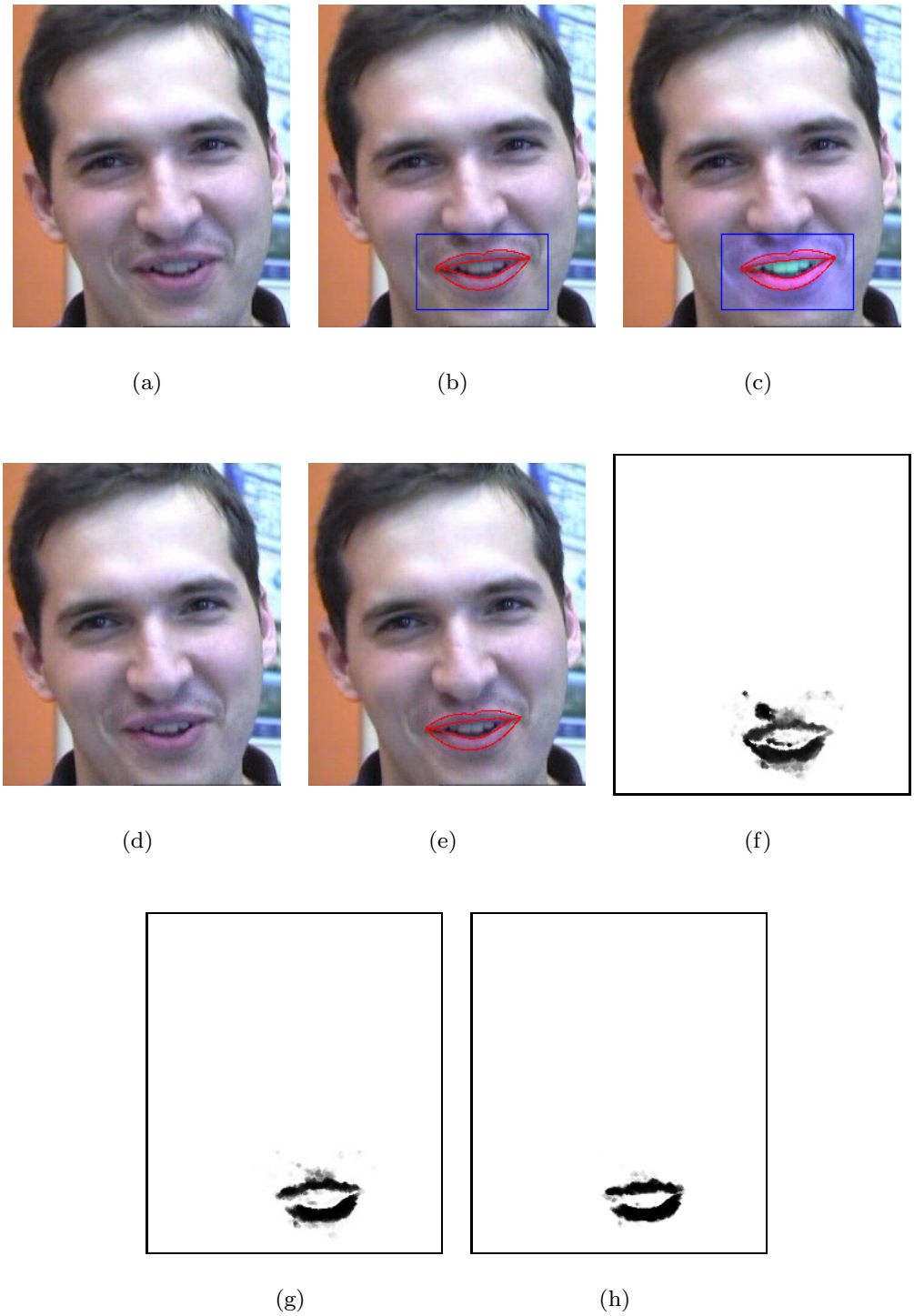


Figure 4.15 : Segmentation floue sous contrainte spatiale : (a) image d'apprentissage I_{n-1} ; (b) région d'intérêt et prototype recalé sur I_{n-1} ; (c) initialisation de 3 classes pour apprentissage ; (d) image de test I_n ; (e) initialisation des classes sur I_n à partir de (c) et en utilisant la pose 3D ; (f)-(h) fonction d'appartenance à la classe "lèvres" pour $\beta = 0$, $\beta = 0.1$ et $\beta = 0.2$, respectivement (valeurs élevées en noir).

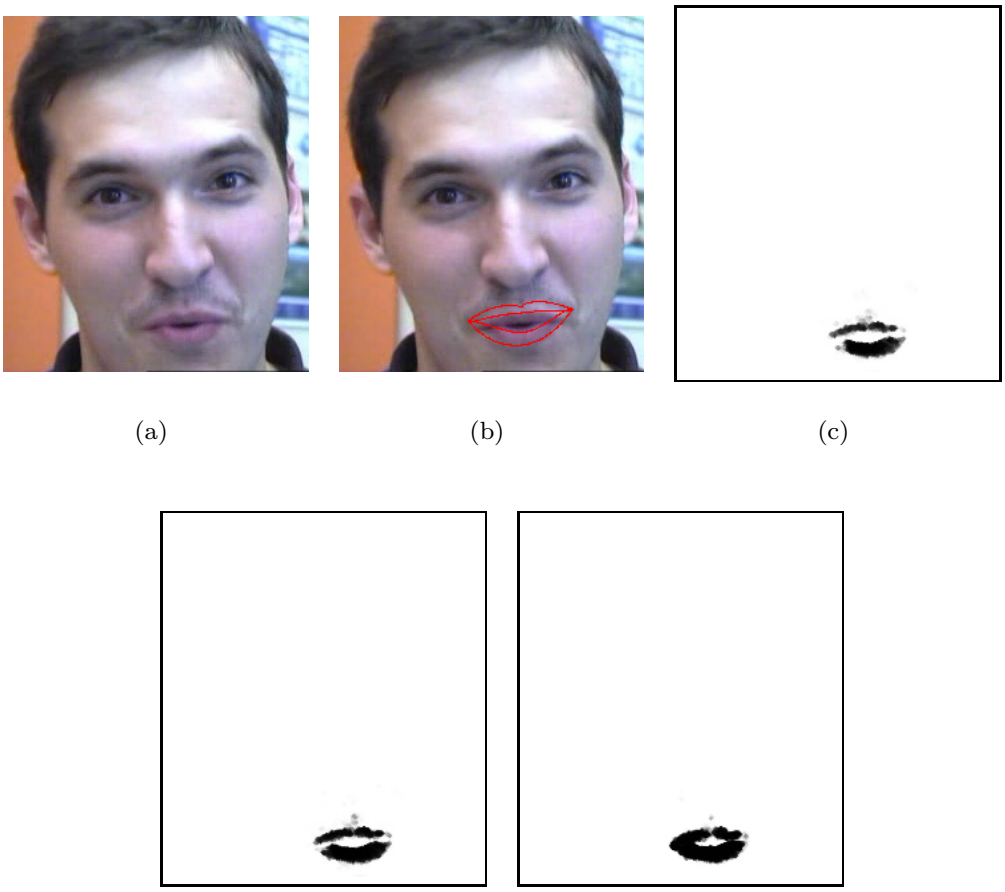


Figure 4.16 : Segmentation floue sous contrainte spatiale à partir de l'apprentissage illustré Figure 4.15 (c) : (a) image de test I_n ; (b) initialisation des classes sur I_n en utilisant la pose 3D ; (c)-(e) fonction d'appartenance à la classe "lèvres" pour $\beta = 0$, $\beta = 0.1$ et $\beta = 0.2$, respectivement (valeurs élevées en noir).

$$E_{\text{texture,bouche}}\left(\mathcal{P}^{(n)}_{\text{bouche}},\, \tau,\, I_n,\, I_{n-1}\right) = \frac{1}{\left|\mathcal{D}^{(n-1)}_{\text{lèvre}}\right|} \iint_{\mathcal{D}^{(n-1)}_{\text{lèvre}}} \sqrt{(I_{n-1} - I_n \circ \tau)^2}\ ,$$

$$E_{\text{segmentation,bouche}}\left(\mathcal{P}^{(n)}_{\text{bouche}},\, \tau,\, I_n,\, I_{n-1}\right) = \frac{1}{\left|\mathcal{D}^{(n-1)}_{\text{lèvre}}\right|} \iint_{\mathcal{D}^{(n-1)}_{\text{lèvre}}} s^{(n)}_{\text{lèvre}} \circ \tau\ ,$$

$$E_{\text{gradient,bouche}}\left(\mathcal{P}^{(n)}_{\text{bouche}},\, \tau,\, I_n,\, I_{n-1}\right) = \sum_{a \in \{\text{s,i}\}} -\frac{1}{\left|\tau(\mathcal{L}^{(n-1)}_{aa})\right|} \int_{\tau(\mathcal{L}^{(n-1)}_{aa})} \|\nabla I_n\|\ ,$$

$$E_{\text{ext,œil ouvert}} = c_{\text{texture,œil ouvert}}\, E_{\text{texture,œil ouvert}} + c_{\text{contraste,œil ouvert}}\, E_{\text{contraste,œil ouvert}}\ ,$$

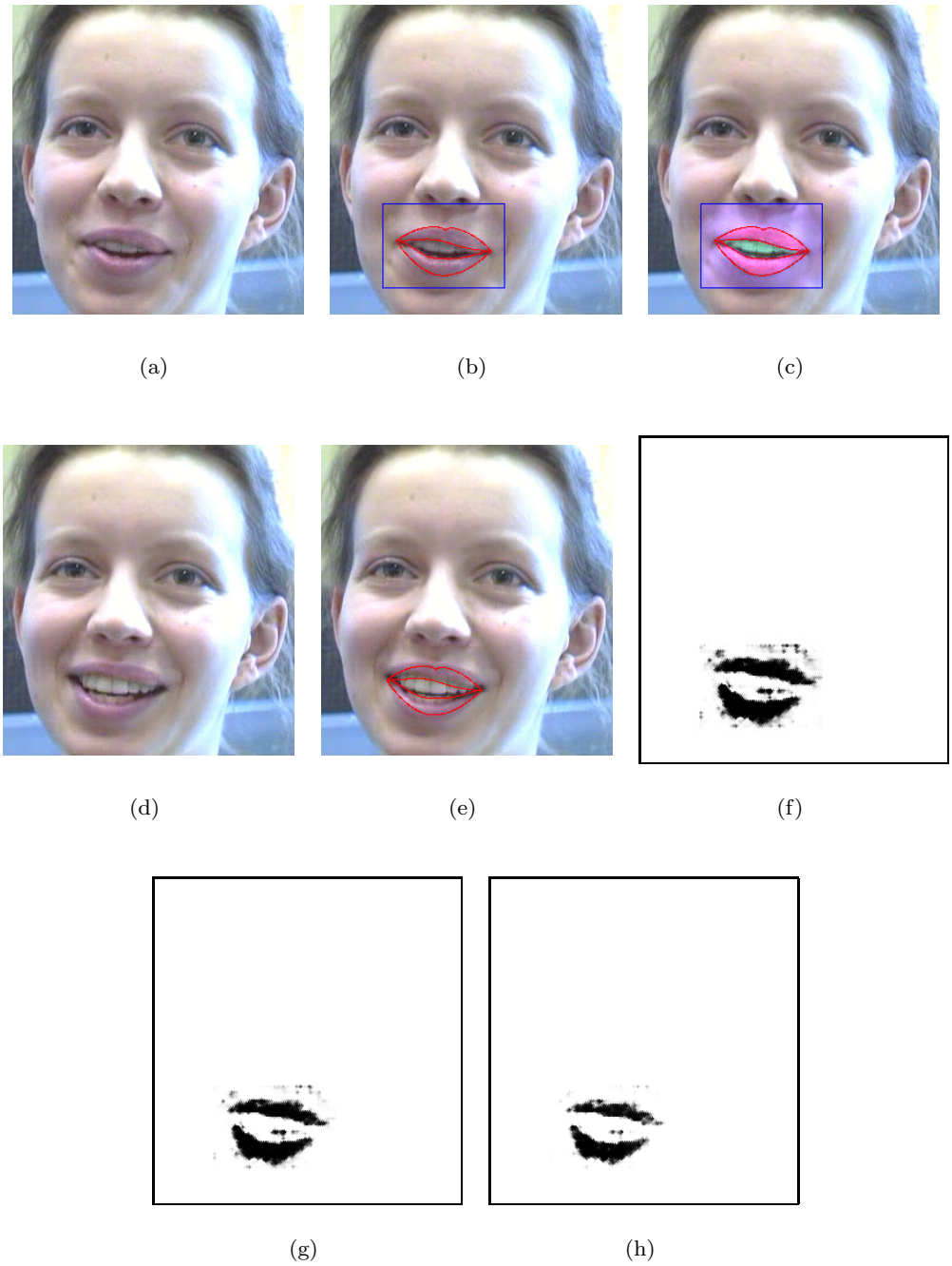


Figure 4.17 : Segmentation floue sous contrainte spatiale : (a) image d'apprentissage I_{n-1} ; (b) région d'intérêt et prototype recalé sur I_{n-1} ; (c) initialisation de 3 classes pour apprentissage ; (d) image de test I_n ; (e) initialisation des classes sur I_n à partir de (c) et en utilisant la pose 3D ; (f)-(h) fonction d'appartenance à la classe "lèvres" pour $\beta = 0$, $\beta = 0.1$ et $\beta = 0.2$, respectivement (valeurs élevées en noir).

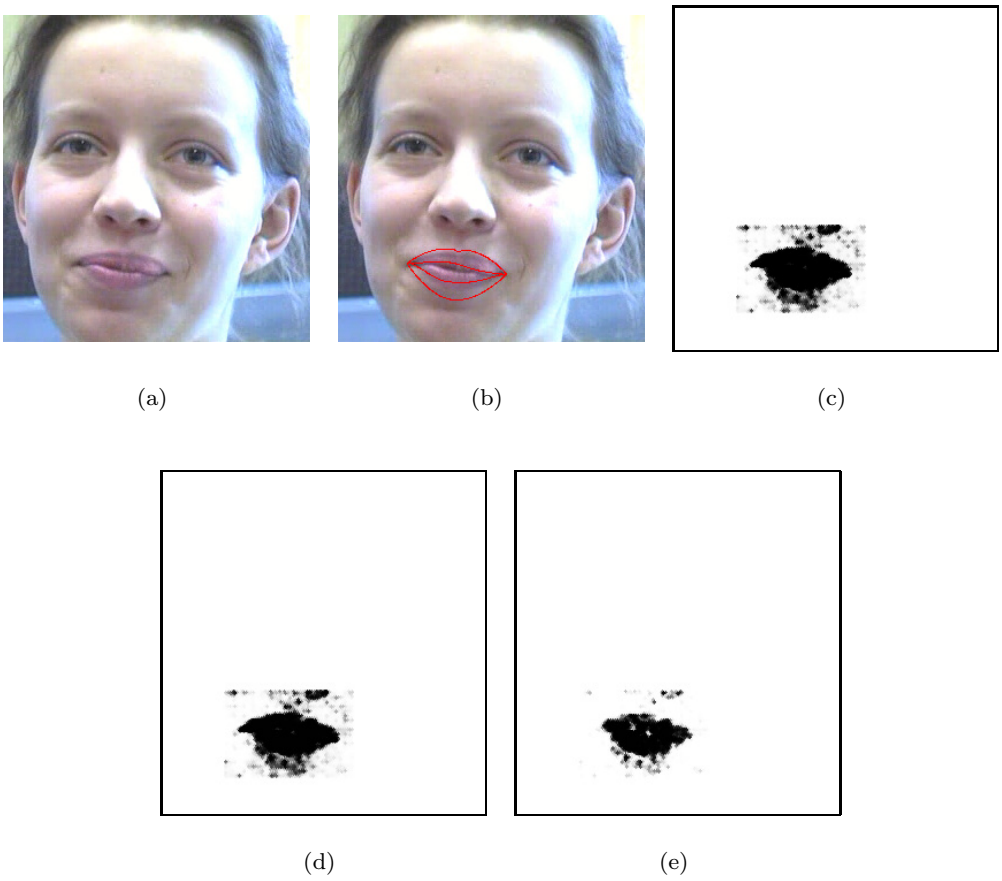


Figure 4.18 : Segmentation floue sous contrainte spatiale à partir de l'apprentissage illustré Figure 4.17 (c) : (a) image de test I_n ; (b) initialisation des classes sur I_n en utilisant la pose 3D ; (c)-(e) fonction d'appartenance à la classe "lèvres" pour $\beta = 0$, $\beta = 0.1$ et $\beta = 0.2$, respectivement (valeurs élevées en noir).

$$E_{\text{texture,œil ouvert}}\left(\mathcal{P}_{\text{œil}}^{(n)},\, \tau,\, I_n,\, I_{n-1}\right) = e_{\text{œil}}^{(n-1)}\, e_{\text{œil}}^{(n)}\, \frac{1}{\left|\mathcal{D}_{\text{blanc}}^{(n-1)} \cup \mathcal{D}_{\text{iris}}^{(n-1)}\right|} \cdot \iint\limits_{\mathcal{D}_{\text{blanc}}^{(n-1)} \cup \mathcal{D}_{\text{iris}}^{(n-1)}} \left|\check{I}_{n-1} - \check{I}_n \circ \tau_{\text{globe oculaire}}\right| \,,$$

$$E_{\text{contraste,œil ouvert}}\left(\mathcal{P}_{\text{œil}}^{(n)},\, \tau,\, I_n,\, I_{n-1}\right) = e_{\text{œil}}^{(n)}\, \frac{\left|\tau_{\text{paupière}}(\mathcal{D}_{\text{blanc}}^{(n-1)})\right|}{\left|\tau_{\text{paupière}}(\mathcal{D}_{\text{iris}}^{(n-1)})\right|}\, \frac{\iint\limits_{\tau_{\text{paupière}}(\mathcal{D}_{\text{iris}}^{(n-1)})} \check{I}_n \circ \tau_{\text{globe oculaire}}}{\iint\limits_{\tau_{\text{paupière}}(\mathcal{D}_{\text{blanc}}^{(n-1)})} \check{I}_n \circ \tau_{\text{globe oculaire}}} \,,$$

où :

- l'exposant indique l'image à laquelle le prototype est associé,

- $e_{\text{œil}}$ spécifie l'état de l'œil, 0 pour l'œil fermé ou 1 pour l'œil ouvert,
- $|\cdot|$ désigne la mesure de longueur ou d'aire pour les courbes ou les surfaces,
- $||\cdot||$ représente la norme euclidienne,
- le chapeau est l'opérateur d'ouverture morphologique en niveaux de gris avec un élément structurant carré,
- ∇ est l'opérateur de gradient de Canny-Deriche,
- $\tau_{\text{paupière}}$ et $\tau_{\text{globe oculaire}}$ désignent deux transformations appliquées à l'œil.

La première modélise le mouvement non rigide de la paupière, tandis que la deuxième donne le mouvement rigide du globe oculaire. Elle est approchée par une translation 2D.

4.5 Suivi de la bouche et des yeux

4.5.1 Algorithme de suivi

Le recalage du prototype sur l'image consiste à minimiser l'énergie totale, définie comme une somme pondérée des deux termes énergétiques introduits dans les sections précédentes :

$$E_{\text{prototype}}(\tau) = c_{\text{int}}\, E_{\text{int}} + c_{\text{ext}}\, E_{\text{ext}} \ .$$

Cette minimisation est effectuée par rapport à la transformation τ déformant le prototype.

Le choix de l'espace de la solution dépend du but recherché. Dans un contexte de segmentation, la minimisation peut être effectuée directement sur les coordonnées des points contrôlant les splines. Mais, quand il s'agit du suivi d'un objet déformable dans une séquence vidéo $(I_n)_n$, on recherche plutôt un minimum dans un espace de transformations τ non rigides paramétrisées. Dans ce dernier cas, l'espace des solutions est réduit à l'espace des paramètres contrôlant τ.

Dans nos expérimentations, la transformation déformant le prototype est modélisée par un polynôme du deuxième degré dont les coefficients sont donc à estimer :

$$\tau(x,y) = \begin{pmatrix} a_0 + a_x\, x + a_y\, y + a_{xx}\, x^2 + a_{xy}\, xy + a_{yy}\, y^2 \\ b_0 + b_x\, x + b_y\, y + b_{xx}\, x^2 + b_{xy}\, xy + b_{yy}\, y^2 \end{pmatrix} \ .$$

Le suivi de l'élément facial dont le prototype associé est $\mathcal{P}$ dans la séquence vidéo consiste en la mise à jour itérative de $\mathcal{P}$ par l'intermédiaire de la transformation τ comme suit. Soit $\widehat{\mathcal{P}}_{n-1}$ le prototype recalé sur l'élément facial, dans l'image I_{n-1}. Les étapes suivantes sont successivement effectuées :

- le mouvement global de la tête est estimé par la procédure décrite dans le chapitre 3 ; la transformation résultant de cette estimation est appliquée à $\widehat{\mathcal{P}}_{n-1}$ pour initialiser le prototype dans l'image I_n (projection 2D/3D/2D) ;

- la configuration ouverte/fermée de l'œil est estimée par l'algorithme décrit dans la section 4.3.2.3 ;

- la transformation optimale $\widehat{\tau}_n$ est estimée par la minimisation de l'énergie totale du prototype, exprimée à base des termes énergétiques précédemment définis, par rapport aux coefficients polynomiaux, via l'algorithme du simplexe ; les itérations sont initialisées avec $a_x = b_y = 1$ et zéro pour tous les autres coefficients. Dans le cas particulier de la configuration "œil fermé", cette étape est ignorée, et on impose au prototype la configuration "œil fermé" ;

- le prototype est mis à jour en appliquant la transformation $\widehat{\tau}_n$ à tous les points caractéristiques et de nouvelles splines sont générées ;

- le modèle élastique est relâché en fixant la position de repos des ressorts à celle correspondant aux points caractéristiques du prototype mis à jour.

L'initialisation des points caractéristiques sur la première image de la séquence vidéo est effectuée de manière interactive.

4.5.2 Résultats

Les algorithmes développés ont été testés sur un corpus de 3 séquences vidéos monoculaires et non calibrées, chacune d'entre elles comportant environ 1000 images d'un visage parlant, avec différentes expressions faciales et mouvements de tête.

En outre, une séquence de calibration, comportant des expressions faciales exagérées a été utilisée afin de régler les paramètres intervenant dans les expressions énergétiques des prototypes. L'objectif d'une telle procédure de calibration est de mettre en place des modèles élastiques souples, capables de supporter de larges déformations et d'établir de manière heuristique l'importance relative des primitives de l'image. Les pondérations des différentes composantes énergétiques obtenues après étude expérimentale sont données au Tableau 4.4. Le rapport optimal entre les composantes interne et externe de l'énergie du prototype est fixé à 2.

	composante d'erreur interne [%]		composante d'erreur externe [%]			
	élastique	symétrie	texture	contour	segmentation	contraste
bouche	33	66	20	40	40	-
œil	50	50	33	-	-	66

Tableau 4.4 : Les pondérations des différentes composantes énergétiques des prototypes.

Tout d'abord, la procédure de suivi automatique de l'iris a été testée et validée sur les séquences de test considérées. Les résultats obtenus montrent la précision et la robustesse de l'algorithme, les erreurs survenant dans moins de $5°/_{\circ\circ}$ des situations.

Appliqué aux mêmes séquences, l'algorithme de recalage des prototypes (yeux et lèvres) démontre une bonne stabilité visuelle dans la grande majorité des situations, comportant des expressions naturelles de visage. Toutefois, des erreurs de recalage peuvent apparaître lorsque l'on rencontre des configurations dégénérées telles que l'absence temporaire des lèvres dans l'image ou de très grandes déformations au niveau du visage.

La Figure 4.19 montre quelques images de prototypes recalés, extraites de deux séquences de test.

4.6 Application à l'animation MPEG-4 d'avatar

La procédure de suivi de primitives de visage à base de prototypes déformables compatibles MPEG-4 trouve une application directe, à travers un schéma d'analyse/synthèse de déformations faciales, dans l'animation d'avatars à partir de séquences vidéos naturelles. L'ensemble d'analyse/synthèse est conçu de la manière suivante (Figure 4.20) :

- un analyseur prend en entrée, d'une part, la succession d'images de la séquence vidéo naturelle et, d'autre part, le modèle 3D de tête et les prototypes associés aux yeux et à la bouche ; à travers l'étape de recalage rigide/déformable, l'analyseur fournit un ensemble de données primaires comportant les paramètres de la pose 3D et les coordonnées des points caractéristiques MPEG-4 correspondant aux yeux et à la bouche ;
- un bloc de formatage transforme les données primaires en FAPs qui sont transmis à un codeur générant du flux MPEG-4 ;
- un synthétiseur utilise ce flux pour animer un modèle de visage compatible MPEG-4.

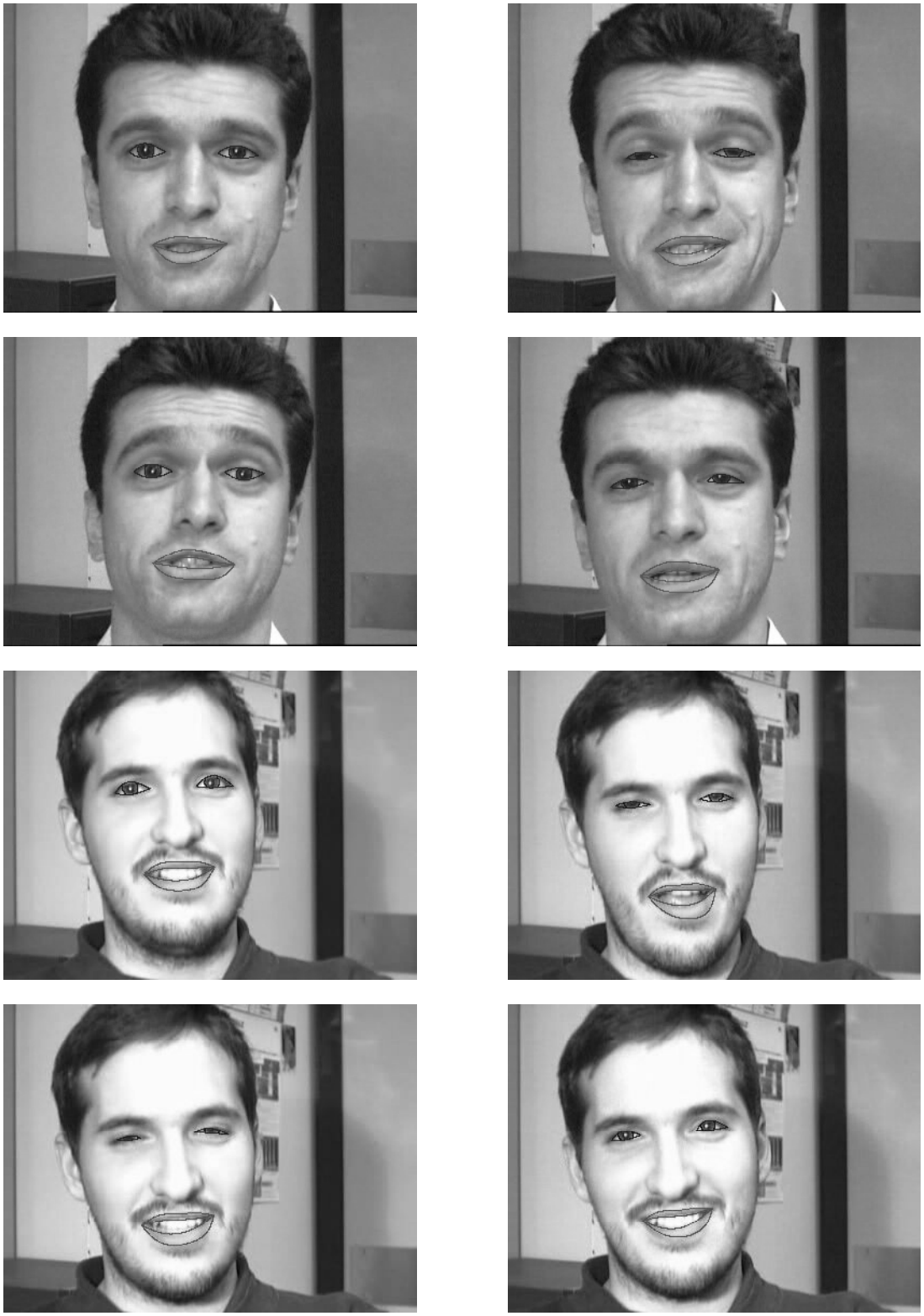


Figure 4.19 : Exemples de prototypes recalés sur les images correspondant à deux séquences de test.

Un extrait d'animation d'avatar correspondant à différentes expressions faciales est illustré Figures 4.21 et 4.22. Mentionnons que, dans notre cas, le flux MPEG-4 contient uniquement les FAPs associés aux yeux et à la bouche. Le réalisme des mouvements du visage de synthèse se trouve ainsi affecté par l'absence d'information de déplacement liée aux autres points caractéristiques de visage qui compléteraient les expressions restituées.

4.7 Conclusion

Nous avons présenté une méthode de recalage à base de prototypes déformables pour le suivi de la bouche et des yeux dans des séquences vidéos monoscopiques contenant un visage parlant.

Les prototypes proposés, conçus de manière à intégrer les descripteurs de visage définis par le standard MPEG-4, sont obtenus dans le cadre d'une représentation souple et précise, à l'aide de B-splines. Ils sont intrinsèquement caractérisés par des contraintes élastiques et de symétrie locale héritées d'une modélisation physique à base de ressorts. Dans le cas particulier de l'œil, deux prototypes ont été proposés, correspondant respectivement à la configuration ouverte/fermée de celui-ci.

Nous avons spécifié les déformations auxquelles sont soumis les prototypes par interaction avec les données images. Typiquement, des primitives de gradient et de texture, combinées à une carte de segmentation floue sous contrainte spatiale sont mises en œuvre, de manière à exploiter les caractéristiques de chaque élément facial considéré.

Nous avons énoncé le principe du suivi des éléments faciaux et détaillé l'algorithme associé. Les contraintes internes et externes des prototypes sont combinées dans un schéma d'optimisation par la méthode du simplexe. L'initialisation robuste des prototypes d'une image à l'autre est effectuée en couplant une procédure de segmentation automatique de l'iris et de détection de la configuration ouverte/fermée de l'œil à l'estimation de la pose 3D de la tête.

Appliqué à des séquences vidéos de visage parlant, acquises dans des conditions réalistes, l'algorithme de suivi proposé démontre une bonne stabilité visuelle dans la grande majorité des situations, comportant des expressions naturelles de visage. Toutefois, des erreurs de recalage peuvent apparaître lorsque l'on rencontre de très grandes déformations.

Enfin, la procédure de suivi de primitives de visage a été intégrée dans un schéma d'analyse/synthèse de déformations faciales, compatible MPEG-4, pour l'animation d'avatars à partir de séquences vidéos naturelles.

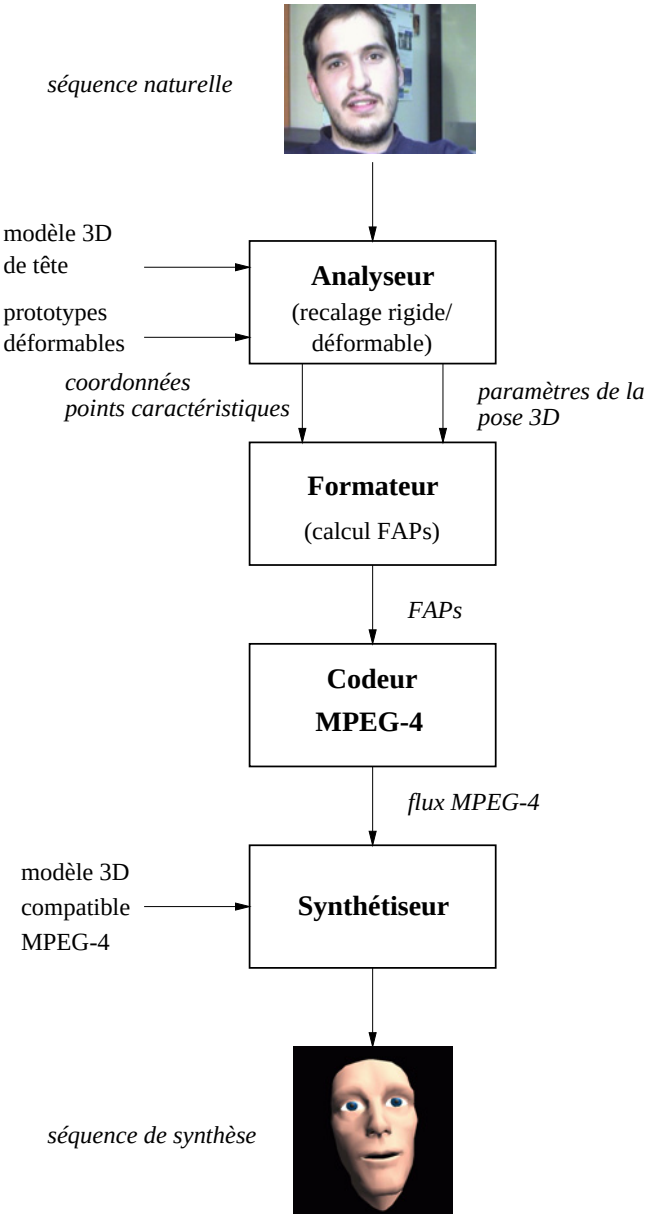


Figure 4.20 : Transformation des données primaires en FAPs à transmettre à un codeur générant un flux MPEG-4.

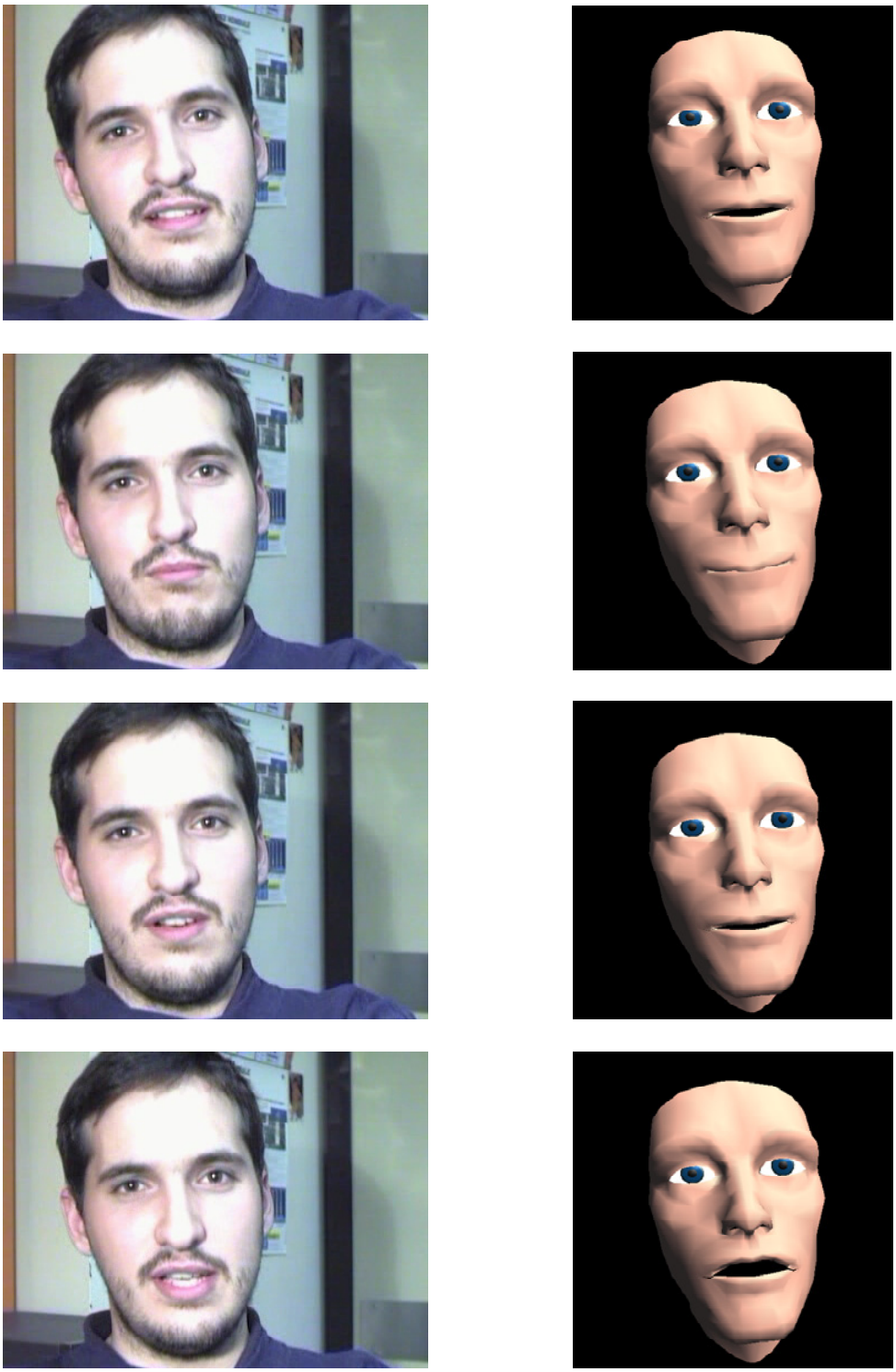


Figure 4.21 : Animation MPEG-4 d'avatar à partir d'une séquence naturelle.

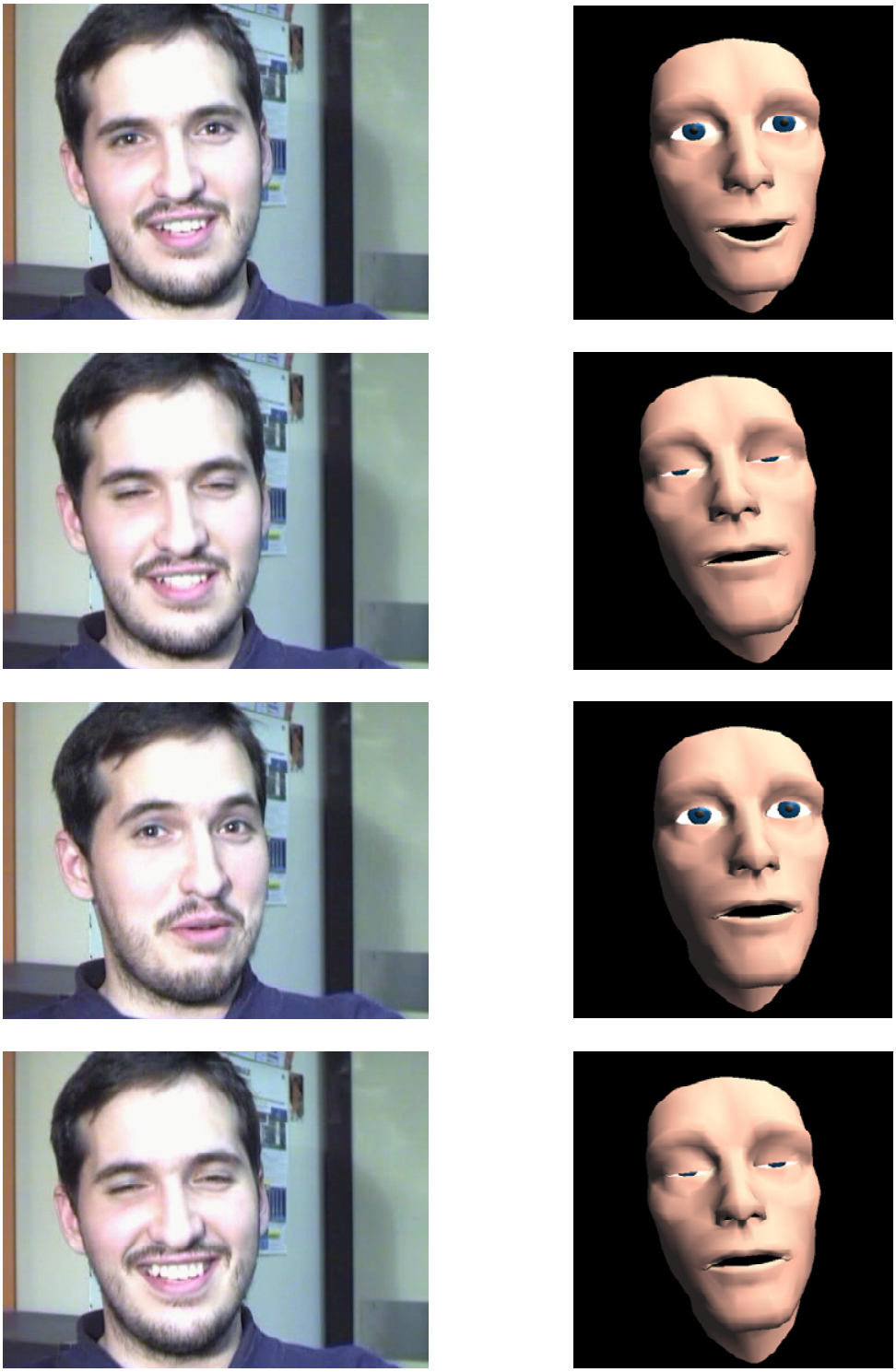


Figure 4.22 : Animation MPEG-4 d'avatar à partir d'une séquence naturelle (suite).

Conclusion et perspectives

Recalage de visage et caractérisation d'expression faciale à partir de séquences vidéos ont suscité ces dernières années de nombreuses recherches dans le cadre d'applications référencées vision telles que l'interaction homme-machine, la reconnaissance de visage ou d'expression faciale et le codage vidéo orienté modèle. Si le système visuel humain permet de localiser spontanément un visage et ses principales composantes, et de reconnaître et différencier les expressions faciales, ces mêmes tâches transposées dans le cadre de la vision par ordinateur restent des sujets ouverts à la recherche. Les principales difficultés à surmonter renvoient à la grande variabilité morphologique du visage et aux déformations locales plus ou moins prononcées liées à la richesse des expressions faciales. En outre, les scènes à analyser, de contenus quelconques, sont acquises par une seule caméra fixe ou mobile, en général non calibrée et dans des conditions d'éclairement *a priori* inconnues et surtout non stabilisées.

Notre recherche sur les approches orientées modèle pour la capture des mouvements de visage a abouti à une méthodologie robuste d'estimation de la pose 3D globale de la tête et de suivi des déformations faciales.

Le premier chapitre de ce mémoire est dédié, dans le cadre de l'état de l'art, à une synthèse des approches de capture des mouvements du visage, rencontrées dans la littérature. Celles-ci sont différenciées en fonction du caractère global ou local du suivi, la discussion de leurs performances et points faibles mettant en perspective les orientations méthodologiques de notre recherche.

L'approche la plus classique mettant en œuvre un modèle d'objet de tête pour analyser des séquences vidéos faciales, comporte deux étapes de difficulté croissante : 1) une adaptation globale de la pose du modèle, en assimilant le mouvement de la tête à un mouvement rigide, 2) une adaptation locale de la forme du modèle, pour prendre en compte les déformations/mouvements des différentes parties du visage. Le chapitre 2 décrit une méthode générique d'estimation de la pose 3D globale de la tête dans des séquences vidéos monoscopiques, non-calibrées, à travers une mise en correspondance de primitives 3D d'un modèle de tête avec des primitives 2D extraites

des images.

Tout d'abord, une méthode de synthèse d'un modèle générique de tête par une approche analytique est présentée et son adéquation à notre problématique est discutée par rapport aux techniques de représentation par maillage.

Les ensembles de primitives 3D du modèle et 2D des images sont ensuite définis. Ces primitives renvoient à des informations de mouvement, au travers du champ de déplacements, et de texture, via la photométrie. Dans l'approche proposée, le champ de déplacements est estimé via un algorithme de flot optique par programmation dynamique orthogonale. Le concept de flot coloré est ensuite dérivé par introduction des informations de texture dans la région de la tête et les primitives définies pour réaliser le recalage 3D/2D du modèle de tête sur les images. Le principe de la mise en correspondance repose sur la minimisation par la méthode du simplexe d'une fonctionnelle d'erreur composite, définie de manière à assurer la propagation d'une image à l'autre, de l'information liée à la pose 3D du modèle, à travers le flot coloré.

Afin de pallier les limitations bien connues d'estimation du flot optique, une compensation en mouvement translationnel dominant est introduite en utilisant une approche de type appariement par bloc.

En dépit d'une bonne précision générale de recalage, les résultats obtenus sur des séquences synthétiques et réelles montrent toutefois des décrochements survenus en présence de mouvements complexes de grande vitesse angulaire générant des auto-occultations ou des occultations occasionnelles.

Le chapitre 3 propose des apports méthodologiques pour s'affranchir des limitations de la méthode d'estimation de la pose 3D de la tête à base de flot coloré. Ainsi, utilisant le champ de déplacements compensé en translation dominante, une technique d'interpolation spatio-temporelle est développé. Elle est fondée sur une modélisation ondulatoire hiérarchique dans le cadre d'une approche physique par groupe de paquets d'onde. Cette interpolation permet de synthétiser des images virtuelles afin de guider un recalage 3D/2D stable et précis en présence de grandes rotations. De plus, pour mieux gérer les auto-occultations de la tête, un critère de visibilité issu des principes d'estimation robuste est introduit.

Ensuite, il est montré comment les cas d'occultation au sens du mouvement peuvent être gérés en considérant une classification fondée sur un critère de mouvement sous contrainte de régularité spatiale, couplée à une analyse de similarité de mouvements à base de modèle paramétrique.

Les simulations numériques sur les séquences calibrées de synthèse montrent que l'estimation est effectuée dans 90% des cas avec une précision de 1.5° pour la rotation de la tête dans le plan

de l'image, de 4° pour l'angle d'azimut, de 8° pour l'angle d'élévation, de 98% (relativement à la taille de la tête) pour les translations et de 97% pour le facteur d'échelle. Appliquée aux séquences réelles, la méthode proposée a permis de supprimer les échecs de recalage précédemment rencontrés, tout en en conservant les bonnes performances pour les configurations simples.

Concernant la capture des mouvements non rigides du visage et des expressions faciales, le chapitre 4 présente une méthode de recalage à base de prototypes déformables pour le suivi de la bouche et des yeux dans des séquences vidéos monoscopiques contenant un visage parlant.

Les prototypes proposés, conçus de manière à intégrer les descripteurs de visage définis par le standard MPEG-4, sont obtenus dans le cadre d'une représentation souple et précise, à l'aide de B-splines. Ils sont intrinsèquement caractérisés par des contraintes élastiques et de symétrie locale héritées d'une modélisation physique à base de ressorts. Dans le cas particulier de l'œil, deux prototypes ont été proposés, correspondant respectivement à la configuration ouverte/fermée de celui-ci.

Les déformations auxquelles sont soumises les prototypes par interaction avec les données images sont ensuite spécifiées. Typiquement, des primitives de gradient et de texture, combinées à une carte de segmentation floue sous contrainte spatiale sont mises en œuvre, de manière à exploiter les caractéristiques de chaque élément facial considéré.

Les contraintes internes et externes des prototypes sont combinées dans un schéma d'optimisation par la méthode du simplexe. L'initialisation robuste des prototypes d'une image à l'autre est effectuée en couplant une procédure de segmentation automatique de l'iris et de détection de la configuration ouverte/fermée de l'œil à l'estimation de la pose 3D de la tête.

Appliqué sur des séquences vidéos de visage parlant, acquises dans des conditions réalistes, l'algorithme de suivi proposé démontre une bonne stabilité visuelle dans la grande majorité des situations, comportant des expressions naturelles de visage. Toutefois, des erreurs de recalage peuvent apparaître lorsque l'on rencontre de très grandes déformations.

La procédure de suivi de primitives de visage a été intégrée dans un schéma d'analyse/synthèse de déformations faciales, compatible MPEG-4, pour l'animation d'avatars à partir de séquences vidéos naturelles.

Les perspectives de ce travail portent aujourd'hui sur l'animation d'un avatar réaliste du point de vue des expressions faciales avec application à un signeur virtuel destiné au monde des sourds et mal-entendants. Dans le contexte du langage des signes, l'information transmise à travers les expressions faciales représente une composante presque aussi importante que l'information portée par le geste. Une analyse-synthèse réaliste d'expression faciale devra enrichir l'ensemble des primitives de visage prises en compte.

De plus, mentionnons que la communication par le langage des signes présente comme particularité l'occultation fréquente du visage par les mains en mouvement. Par conséquent, un suivi robuste d'expression faciale devra prendre en compte une analyse d'occultation au sens du mouvement.

Liste des publications associées

Communications dans des congrès internationaux avec actes et comité de lecture

M. Malciu, F. Prêteux, "MPEG-4 compliant tracking of facial features in video sequences", *Proceedings EUROIMAGE International Conference on Augmented, Virtual Enviroments and Three-Dimensional Imaging ICAV3D'01*, pp. 108-111, Mykonos, Greece, 2001.

F. Prêteux, C. Fetita, M. Malciu, M. Preda, "Advanced methods for 3D object representation and animation. Application to medical imaging and virtual humanoids", *Proceedings of the International Conference Communications 2000* (organisé par l'Académie Technique Militaire, Université "Politehnica" de Bucarest et IEEE - section roumaine), pp. 38-49, Bucarest, 2000.

M. Malciu, F. Prêteux, "Tracking facial features in video sequences using a deformable model-based approach", *Proceedings SPIE Conference on Mathematical Modeling, Estimation and Imaging*, vol. SPIE 4121, pp. 51-62, San Diego, CA, 2000.

M. Malciu, F. Prêteux, "A robust model-based approach for 3D head tracking in video sequences", *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition FG'2000*, pp. 169-174, Grenoble, France, 2000.

M. Malciu, F. Prêteux, V.Buzuloiu, "3D global head pose estimation : A robust approach", *Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging IWSNHC3DI'99*, pp. 79-82, Santorini, Greece, 1999.

F. Prêteux, M. Malciu, "Model-based head tracking and 3D pose estimation", *Proceedings SPIE Conference on Mathematical Modeling and Estimation Techniques in Computer Vision*, vol. SPIE 3457, pp. 94-108, San Diego, CA, 1998.

F. Prêteux, S. Curila, M. Malciu, "Active 3D model-based registration", *Proceedings SPIE Conference on Nonlinear Image Processing IX - IS&T / SPIE Symposium on Electronic Imaging, Science and Technology '98*, vol. SPIE 3304, pp. 186-196, San Jose, CA, 1998.

Communications dans des congrès nationaux avec actes et comité de lecture

M. Malciu, L.-T. Nessi, F. Preteux, "Pose 3D du visage dans des séquences vidéos : estimation robuste par modèle d'objet", *12ème Congrès Francophone AFRIF-AFIA Reconnaissance des Formes et Intelligence Artificielle RFIA'2000*, vol. 1, pp. 24-36, Paris, France, 2000.

Bibliographie

[Abrantes97] G. Abrantes, F. Pereira, "An MPEG-4 SNHC compatible implementation of a 3D facial animation system", *Proceedings of the International Workshop on Synthetic Natural Hybrid Coding and Three Dimensional Imaging IWSNHC3DI'97*, Rhodos, Greece, 1997.

[Ahlberg01] J. Ahlberg, "Using the active appearance algorithm for face and facial feature tracking", *Proceedings of the 2nd International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems RATFFG-RTS*, pp. 68-72, Vancouver, Canada, July 2001.

[Aizawa95] K. Aizawa, T. S. Huang, "Model-based image coding : advanced video coding techniques for very low bit-rate applications", *Proceedings of the IEEE*, vol. 83, no. 2, pp. 259-271, February 1995.

[Ambroise97] C. Ambroise, M. Dang, G. Govaert, "Clustering of spatial data by the EM algorithm", in A. Soares, J. Gomez-Hernandez, R. Froidevaux, eds., *Geostatistics for Environmnental Applications*, pp. 493-504, Kluwer Academic Publisher, 1997.

[Ambroise98] C. Ambroise, G. Govaert, *Convergence of an EM-type algorithm for spatial clustering* Pattern Recognition Letters vol. 19, no. 10, pp.919-927, August 1998.

[Andreff01] N. Andreff; R. Horaud, B. Espiau, "Robot hand-eye calibration using structure-from-motion", *International Journal of Robotics Research*, vol. 20, no. 3, pp.228-248, March 2001.

[Antoszczyszyn98] P. M. Antoszczyszyn, J. M. Hannah, P. M. Grant, "Reliable tracking of facial features in semantic-based video coding", *IEE Proceedings on Vision Image and Signal Processing*, vol. 145, no. 4, pp. 257-263, August 1998.

[Azarbayejani95] A. Azarbayejani, A. Pentland, "Recursive estimation of motion, structure and focal length", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 562–575, June 1995.

[Barron94] J. L. Barron, D.J. Fleet, S. S. Beauchemin, "Performance of optical flow techniques", *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43-77, February 1994.

[Bartels87] R. H. Bartels, J. C. Beatty, *An introduction to splines for use in computer graphics and geometric modeling*, Morgan Kaufman, Los Altos, CA, 1987.

[Basu96] S. Basu, I. Essa, A. Pentland, "Motion regularization for model-based head tracking", *Proceedings of the IEEE International Conference on Pattern Recognition ICPR'96*, vol. 3, pp. 611-616, Vienna, Austria, September, 1996.

[Beier92] T. Beier. S. Neely, "Feature-based image metamorphosis", *Computer Graphics*, vol. 26, no. 2, pp. 35–42, 1992.

[Black93] M. Black, P. Anandan, "A framework for the robust estimation of the optical flow", *Proceedings of the IEEE International Conference on Computer Vision ICCV'93*, pp. 231-236, Berlin, Germany, May 1993.

[Black95] M. J. Black, Y. Yacoob, "Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion", *Proceedings of the IEEE International Conference on Computer Vision ICCV'95*, pp. 374–381, Boston, MA, June 1995.

[Blake92] A. Blake, A.Yuille, eds., *Active Vision*, MIT Press, 1992.

[Bozdagi94] G. Bozdagi, A. M. Tekalp, L. Onural, "3-D motion estimation and wireframe model adaptation including photometric effects for model-based coding of facial image sequences", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 246-257, June 1994.

[Bregler94] C. Bregler, Y. Konig, "Eigenlips for robust speech recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'94*, vol. 2, pp. 669-672, Adelaide, Australia, 1994.

[Broida90] T. J. Broida, S. Chandrashekhar, R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 4, pp. 639-656, July 1990.

[Brunelli93] R. Brunelli, T. Poggio "Face recognition : Features versus templates", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 15, no. 10, pp. 1042–1062, October 1993.

[Brunelli94] R. Brunelli., "Estimation of pose and illuminant direction for face processing", *A. I. Memo* no. 1499, Massachusetts Institute of Technology, 1994.

[Burt84] P. J. Burt, "The pyramid as a structure for efficient computation", in A. Rosenfeld, ed., *Multiresolution image processing and analysis*, pp. 6-35, Springer-Verlag, Berlin/New-York, 1984.

[Canny86] J. Canny, "A computational approach to edge detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, November 1986.

[Chellappa95] R. Chellappa, C. Wilson, S. Sirohey, "Human and machine recognition of faces: A survey," *Proceedings of the IEEE*, vol. 83, no. 5, pp. 705–740, May 1995.

[Chiou97] G. I. Chiou, Jenq-Neng Hwang, "Lipreading from color video", *IEEE Transactions on Image Processing*, vol. 6, no. 8, pp. 1192-1195, August 1997.

[Choi94] C. S. Choi, K. Aizawa, H. Harashima, T. Takebe, "Analysis and synthesis of facial image sequences in model-based image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 257-275, June 1994.

[Colmenarez97] A. Colmenarez, R. Lopez, and T. Huang, "3d model-based head tracking", *Proceedings of the SPIE International Conference on Visual Communications and Image Processing VCIP'97*, vol. SPIE 3024, pp. 426–434, San Jose, CA, February 1997.

[Cootes94] T. F. Cootes, A. Hill, C. J. Taylor, J. Haslam, "The use of active shape models for locating structures in medical images", *Image and Vision Computing*, vol. 12, no. 6, pp. 355-366, July 1994.

[Cootes95] T. F. Cootes, C. J. Taylor, D. H. Cooper, J. Graham, "Active shape models - their training and application", *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38-59, January 1995.

[Cootes98] T. F. Cootes, G. J. Edwards, C. J. Taylor. "Active appearance models" *Proceedings of the 5th European Conference on Computer Vision ECCV'98*, pp. 484-498, Freiburg, Germany, 1998.

[Cordea01] M. D. Cordea, E. M. Petriu, N. D. Georganas, D. C. Petriu, T. E. Whalen, T.E. "3D head pose recovery for interactive virtual reality avatars", *Proceedings of the IEEE Conference on Instrumentation and Measurement Technology Conference IMTC'2001*, pp. 72-77, Budapest, Hungary, May 2001.

[Cox77] M. G. Cox, "The incorporation of boundary conditions in spline approximation problems", *NPL Reeport*, NAC 80, 1977.

[Crowley95] J. L. Crowley, J. Coutaz, "Vision for man machine interaction", *Proceedings of the IFIP Working Conference on Engineering for Human-Computer Interaction*, pp. 28-45, Grand Targhee Resort, WY, August 1995.

[Curila99] S. Curila, M. Curila, T. Zaharia, G. Mozelle, F. Prêteux, "A new prediction scheme for geometry coding of 3D meshes within the MPEG-4 framework", *Proceedings of the SPIE International Conference on Nonlinear Image Processing*, vol. SPIE 3646, pp. 240-250, San Jose, CA, 1999.

[Debevec96] P. E. Debevec, C. J. Taylor, J. Malik, "Modeling and rendering architecture from photographs : A hybrid geometry- and image-based approach", *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques SIGGRAPH'96*, pp. 11-20, New Orleans, LA, 1996.

[DeBoor78] C. De Boor. *A Practical Guide to Splines*, Springer-Verlag, New York, USA, 1978.

[DeCarlo96] D. DeCarlo, D. Metaxas, "The integration of optical flow and deformable models with applications to human face shape and motion estimation", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition CVPR '96*, pp. 231-238, San Francisco, CA, June1996.

[DeMenthon96] D. F. Dementhon, L. S. Davis, "Model-based object pose in 25 lines of code", *International Journal of Computer Vision*, vol. 15, no. 1-2, pp. 123-141, June 1995.

[Dempster77] A. P. Dempster, N. M. Laird, D. B. Rubin. "Maximum likelihood for incomplete data via the EM algorithm", *Journal of the Royal Statistical Society*, series B, vol. 39, pp. 1–38, 1977.

[Eisert98] P. Eisert, B. Girod, "Analyzing Facial Expression for Virtual Conferencing", *IEEE Transactions on Computer Graphics and Applications*, vol. 18, no. 5, pp. 70-78, September 1998.

[Ekman77] P. Ekman W. V. Friesen, "Facial action coding system", *Consulting Psychologist Press*, Palo Alto, CA, 1977.

[Enkelmann86] W. Enkelmann, "Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences", *Proceedings of the IEEE Work-*

shop on Motion : Representation and Analysis, pp. 81-87, Charleston, SC, May 1986.

[Essa97] I. Essa, A. Pentland, "Coding, analysis, interpretation, and recognition of facial expressions", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 757-763, July 1997.

[Faugeras92] O. Faugeras, "From geometry to variational calculus : Theory and applications of three-dimensional vision", *Proceedings of the IEEE Workshop on Computer Vision for Virtual Reality Based Human Communicationsin CVVRHC'98*, pp. 52-71, Bombay, India, January 1998.

[Fleet90] D. Fleet, A. Jepson, "Computation of component image velocity from local phase information", *International Journal of Computer Vision*, vol. 5, no. 1, pp. 77–104, August 1990.

[Fukuhara93] T. Fukuhara, T. Murakami, "3-D motion estimation of human head for model-based image coding", *IEE Proceedings on Communications, Speech and Vision*, vol. 140, no. 1, pp. 26 -35, February 1993.

[Gee94] A. Gee, R. Cipolla. "Determining the gaze of faces in images", *Image and Vision Computing*, vol. 12, no. 10, pp. 639-647, December 1994.

[Glazer87] F. C. Glazer, "Hierarchical gradient-based motion detection", *DARPA Proceedings of Image Understanding Workshop*, pp. 733-748, Los Angeles, CA, February 1997.

[Gleicher97] M. Gleicher. "Projective registration with difference decomposition", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition CVPR'97*, pp. 331-337, San Juan, Puerto Rico, June 1997.

[Grammalidis00] N. Grammalidis, N. Sarris, C. Varzokas, M. G. Strintzis, M.G., "Generation of 3-D head models from multiple images using ellipsoid approximation for the rear part", *Proceedings of the IEEE International Conference on Image Processing ICIP'2000*, vol. 1, pp. 284-287, Vancouver, Canada, September 2000.

[Haslam94] J. Haslam, C. J. Taylor, T. F. Cootes. "A probabalistic fitness measure for deformable template models", *Proceedings of the 5th British Machine Vision Conference*, vol. 1, pp. 33-42, York, UK, September 1994.

[Hathaway86] R. J. Hathaway, "Another interpretation of the EM algorithm for mixture distributions", *Journal of Statistics and Probability Letters*, vol. 4, pp. 53-56, 1986.

[Hay66] J. C. Hay, "Optical motions and space perceptions : An extension of Gibson's analysis", *Psychological Review*, vol. 73, no. 6, pp. 550-565, 1996.

[Horaud98] R. Horaud, F. Dornaika, B. Espiau, "Visually guided object grasping", *IEEE-Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 525-532, August 1998.

[Horprasert96] T. Horprasert, Y. Yacoob, L.S Davis, "Computing 3-D head orinetation from a monocular image sequence", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 242-247, Killington, VT, October 1996.

[Horn81] B. K. P. Horn, B. G. Schunk, "Determining optic flow", *Artificial Intelligence*, vol. 17, pp. 185–204, 1981.

[Huang93] W. C. Huang, D. B. Goldgof, "Adaptive-size meshes for rigid and nonrigid shape analysis and synthesis", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 15, no. 6, pp. 611-616, June 1993.

[Hubert81] P. J. Hubert, *Robust Statistics*, Wiley, New York, 1981.

[Hughes92] J. F. Hughes, "Scheduled Fourier volume morphing", *Proceedings of the ACM International Conference on Computer Graphics and Interactive Techniques SIGGRAPH'92*, pp. 43-46, Chicago, IL, July 1992.

[Jebara99] T. Jebara, A. Azarbayejani, A. Pentland, "3D Structure from 2D Motion", *IEEE Signal Processing Magazine*, vol. 16, no. 3, pp. 66-84, May 1999.

[Kampmann97] M. Kampmann, J. Ostermann, "Automatic adaptation of a face model in a layered coder with an object-based analysis-synthesis layer and a knowledge-based layer", *Signal Processing : Image Communications*, vol. 9, no. 3, pp. 201-220, March 1997.

[Kass87] M. Kass, A. Witkin, D. Terzopoulos, "Snakes : Active contour models", *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321-231, 1987.

[Kobayashi97] H. Kobayashi, F. Hara, "Facial interaction between animated 3D face robot and human beings", *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Computational Cybernetics and Simulation*, vol. 4, pp. 3732-3737, Orlando, FL, October 1997.

[Koller93] D. Koller, K. Daniilidis, H.-H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes", *International Journal of Computer Vision*, vol. 10, no. 3, pp. 257-281, June 1993.

[Kollnig97] H. Kollnig, H.-H. Nagel, "3D pose estimation by directly matching polyhedral models to gray value gradients", *International Journal of Computer Vision*, vol. 23, no. 3, pp. 283-302, June 1997.

[LaCascia98] M. LaCascia, J. Isidoro, S. Sclaroff, "Head tracking via robust registration in texture map images", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition CVPR'98*, pp. 508-514, Santa Barbara, CA, June 1998.

[Leroy95] B. Leroy, I. L. Herlin, "Un modèle déformable paramétrique pour la reconnaissance de visages et le suivi du mouvement des lèvres", *Proceedings GRETSI'95*, pp. 701-704, Juan les Pins, Septembre 1995.

[Li93] H. Li, P. Roivainen, R. Forchheimer, "3-D motion estimation in model-based facial image coding", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 545-555, June 1993.

[Lien00] J. J.-J. Lien, T. Kanade, J. F. Cohn, C.-C. Li, "Detection, tracking, and classification of action units in facial expression", *Robotics and Autonomous Systems*, vol. 31, no. 3, pp. 131-146, May 2000.

[Longuet-Higgins80] H. C. Longuet-Higgins, K. Prazdny, "The interpretation of a moving retinal image", *Proceedings of Royal Society London*, vol. B 208, pp. 385-397, 1980.

[Lucas81] B. D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision", *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, Vancouver, Canada, 1981.

[Lucey00] S. Lucey, S. Sridharan, V. Chandran, "Initialised eigenlip estimator for fast lip tracking using linear regression", *Proceedings of the IEEE International Conference on Pattern Recognition ICPR'2000*, vol. 3, pp. 178-181, Barcelona, Spain, September 2000.

[Luettin96] J. Luettin, N. A. Tracker, S. W. Beet. "Active shape models for visual speech feature extraction", *Electronic Systems Group Report* no. 95/44, University of Sheffield, UK, 1995.

[Machin96] D. Machin, "Real-time facial motion analysis for virtual teleconferencing", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 340-344, Killington, VT, October 1996.

[Marr76] D. Marr, T. Poggio, "Cooperative computation of stereo disparity", *Science*. vol. 194, no. 4262, pp. 283-287, October 1976.

[Metaxas93] D. Metaxas, S. J. Dickinson, "Integration of quantitative and qualitative techniques for deformable model fitting from orthographic, perspective, and stereo projections", *Proceedings of the IEEE International Conference on Computer Vision ICCV'93*, pp. 641-649, Berlin, Germany, May 1993.

[Moghaddam97] B. Moghaddam, A. Pentland, "Probabilistic visual learning for object representation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696-710, July 1997.

[Moses95] Y. Moses, D. Reynard, A. Blake, "Determining facial expressions in real time", *Proceedings of the IEEE International Conference on Computer Vision ICCV'95*, pp. 296-301, Cambridge, MA, June 1995.

[MPEG-4] *Audio and video object coding*, MPEG-4 ISO/IEC 14496-1.

[Nagel86] H.-H. Nagel, W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 5, pp. 565–593, September 1986.

[Nelder65] J. A. Nelder, R. Mead, "A simplex method for function optimization", *Computer Journal*, vol. 7, no. 4, pp. 308-313, 1965.

[Odobez95] J.-M. Odobez, P. Bouthemy, "Robust multiresolution estimation of parametric motion models", *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348-365, December 1995.

[Oliensis91] J. Oliensis, J. I. Thomas, "Incorporating motion error in multi-frame structure from motion", *Proceedings of the IEEE Workshop on Visual Motion*, pp. 8-13, Princeton, NJ, October 1991.

[Pramadihanto98] D. Pramadihanto, Y. Iwai, M. Yachida, "A flexible feature matching for automatic face and facial feature points detection", *Proceedings of the IEEE International Conference on Pattern Recognition ICPR'98*, vol. 1, pp. 92-95, Brisbane, Australia, August 1998.

[Pardas00] M. Pardas, "Extraction and tracking of the eyelids", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'00*, vol. 4, pp. 2357-2360, Istanbul, Turkey, June 2000.

[Pentland91] A. Pentland, "Photometric motion", *IEEE Transactions on Pattern Analysis and Machine Inteligence*, vol. 13, no. 9, pp. 879-890, September 1991.

[Petajan96] E. Petajan, H. P. Graf, "Robust face feature analysis for automatic speechreading and character animation", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 357-362, Killington, VT, October 1996.

[Press98] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical recipes in C: the art of scientific computing*, Cambridge University Press, 1998.

[Quénot92] G. M. Quénot, "The ‹‹orthogonal algorithm››for optical flow detection using dynamic programming", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'92*, vol. 3, pp.249-252, San Francisco, CA, March 1992.

[Quénot96] G. M. Quénot, "Computation of optical flow using dynamic programming", *IAPR Workshop on Machine Vision Applications*, pp. 249-252, Tokyo, Japan, November 1996.

[Quénot97] G. M. Quénot, "Computation of optical flow using dynamic programming and applications", *Computer Vision and Pattern Recognition CVPR'97*, Demo Program p. 5, San Juan, Puerto Rico, June 1997.

[Reinders95] M. J. T. Reinders, P. J. L. van Beek, B. Sankur, J. C. A. van der Lubbe, "Facial feature localization and adaptation of a generic face model for model-based coding", *Signal Processing : Image Communication*, vol. 7, no. 1 pp. 57-74, March 1995.

[Rohr97] K. Rohr, "Human movement analysis based on explicit motion models", in M. Shah, R. Jain, eds., *Motion-based recognition*, pp. 171-198, Computational Imaging and Vision Series, Vol. 9, Kluwer Academic Publishers, 1997.

[Sanchez97] M. U. Ramos-Sanchez, J. Matas, J. Kittler, "Statistical chromaticity-based lip tracking with B-splines", *IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'97*, vol. 4, pp. 2973-2976, Munich, Germany, April 1997.

[Schubert00] A. Schubert, "Detection and tracking of facial features in real time using a synergistic approach of spatio-temporal models and generalized Hough-transform techniques", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition FG'2000*, pp. 116-121, Grenoble, France, March 2000.

[Serra94] J. Serra, *Mathematical morphology and its applications to image processing*, Kluwer Academic Publishers, 1994.

[Simoncelli99] E. P. Simoncelli, "Bayesian multi–scale differential optical flow" in B. Jähne, H. Haussecker, P. Geissler, eds., *Handbook of computer vision and applications*, pp. 397-422, Academic Press, 1998.

[Slama80] C. C. Slama, ed., *Manual of photogrammetry*, 4th ed., Falls Church, VA : American Society of Photogrammetry and Remote Sensing, 1980.

[Soatto93] S. Soatto, P. Perona, R. Frezza, G. Picci, "Recursive motion and structure estimation with complete error characterization", *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition CVPR'93*, pp. 428-433, New York, NY, June 1993.

[Sobottka96] J. Sobottka, I. Pitas, "Segmentation and tracking of faces in color images", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 236-241, Killington, VT, October 1996.

[Stiefelhagen98] R. Stiefelhagen, J. Yang, A. Waibel, "Towards tracking interaction between people", *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments*, pp. 123–127, Stanford, CA, March 1998.

[Strinzis99] M. G. Strintzis, S. Malassiotis, "Object-based coding of stereoscopic and 3D image sequences", *IEEE Signal Processing Magazine* vol. 16, no. 3, pp. 14-28, May 1999.

[Strom99] J. Strom, T. Jebara, S. Basu, A. Pentland, " Real time tracking and modeling of faces : An EKF-based analysis by synthesis approach", *Technical Report* TR506, Massachusetts Institute of Technology, Media Laboratory, 1999.

[Sum01] K. L. Sum, W. H. Lau, S. H. Leung, A. W. C. Liew, K. W. Tse, "A new optimization procedure for extracting the point-based lip contour using active shape model", *Proceedings of the IEEE International Conference on Accoustics, Speech, and Signal Processing ICASSP'2001*, vol 3, pp. 1485 -1488, Salt Lake City, Utah, May 2001.

[Tarel98] J. P. Tarel, H. Civi, D. B. Cooper, "Pose estimation of free-form 3D objects without point matching using algebraic surface models", *IEEE Workshop on Model Based 3D Image Analysis MB3IA'98*, Bombay, India, January 1998.

[Taubin98] G. Taubin, J. Rossignac, "Geometric compression through topological surgery", *ACM Transactions on Graphics*, vol. 17, no. 2, pp. 84-115, April 1998.

[Tian99] Y. Tian, T. Kanade, J. Cohn, "Multi–state based facial feature tracking and detection", *Technical Report* CMU–RI–TR–99–18, Robotics Institute, Carnegie Mellon University, August, 1999.

[Terzopoulos93] D. Terzopoulos, K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15. no. 6, pp. 569-579, June 1993.

[Tomasi92] C. Tomasi, T. Kanade, "Shape and motion from image streams under orthography: a factorization method", *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137-154, November 1992.

[Torre2000] F. De la Torre, Y. Yacoob, L. Davis, "A probabilistic framework for rigid and non-rigid appearance based tracking and recognition", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition FG'2000*, pp. 491-498, Grenoble, France, March 2000.

[Turc91] M. A. Turk, A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-96, 1991.

[Verri89] A. Verri, T. Poggio, "Motion field and optical flow : Qualitative properties", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 490–498, May 1989.

[Viola95] P. Viola, W. Wells, "Alignment by maximization of mutual information", *Proceedings of the IEEE International Conference on Computer Vision ICCV'95*, pp. 16–23, Cambridge, MA, June 1995.

[WangM98] M. Wang, Y. Iwai, M. Yachida, "Recognizing degree of continuous facial expression change", *Proceedings of the IEEE International Conference on Pattern Recognition ICPR'98*, vol. 2, pp. 1188 -1190, Brisbane, Australia, August 1998.

[WangR97] R.-S. Wang, Y. Wang, "Facial feature extraction and tracking in video sequences", *IEEE First Workshop on Multimedia Signal Processing*, pp. 233 -238, 1997.

[WangY94] Y. Wang, O. Lee, "Active mesh - a feature seeking and tracking image sequence representation scheme", *IEEE Transactions on Image Processing*,vol. 3, no. 5, pp. 610 -624, September 1994 .

[Welsh91] B. Welsh, *Model-based coding of images*, PhD disertation, British Telecom Research Laboratory, January 1991.

[Wu83] C. F. Wu , "On the Convergence properties of the EM algorithm", *The Annals of Statistics*, vol. 11, no. 1, pp. 95-103, 1983.

[Xu98] Xu-Yanjun, Du-Limin, Hou-Ziqiang, "A novel lip localization method based on shiftable wavelets transform", *Proceedings. of the IEEE International Conference on Signal Processing ICSP'98*, vol. 2, pp. 1029-1032, Beijing, China, October 1998.

[Yang99] M.-H. Yang, N. Ahuja, D. Kriegman, "Face detection using a mixture of factor analyzers", *Proceedings of the IEEE International Conference on Image Processing ICIP'99*, vol. 3, pp. 612-616, Kobe, Japan, October 1999.

[Yuille92] A. L. Yuille, P. W. Hallinan, D. S. Cohen, "Feature extraction from faces using deformable templates", *International Journal of Computer Vision*, vol.8, no.2, pp. 99-111, August 1992.

[Zaharia01] T. Zaharia, F. Prêteux, "Parametric motion models for video content description within the MPEG-7 framework", *Proceedings of the SPIE International Conference on Nonlinear Image Processing and Pattern Analysis*, vol. SPIE 4304, pp. 118-132, San Jose, CA, January 2001.

[Zelinsky96] A. Zelinsky, J. Heinzmann, "Human-robot interaction using facial gesture recognition", *Proceedings of the IEEE International Workshop on Robot and Human Communication RO-MAN'96*, pp. 256 -261, Tsukuba, Japan, November 1996.

[Zhang.L98] L. Zhang, "Automatic adaptation of a face model using action units for semantic coding of videophone sequences", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 6, pp. 781-795, October 1998.

[Zhang.Y00] Ye-Zhang, C. Kambhamettu, "Robust 3D head tracking under partial occlusion", *Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition FG'2000*, pp. 176-182, Grenoble, France, March 2000.

Recalage de visage et caractérisation d'expression faciale à partir de séquences vidéos ont suscité ces dernières années de nombreuses recherches dans le cadre d'applications référencées vision telles que l'interaction homme-machine, la reconnaissance de visage ou d'expression faciale et le codage vidéo orienté modèle. Si le système visuel humain permet de localiser spontanément un visage et ses principales composantes, et de reconnaître et différencier les expressions faciales, ces mêmes tâches transposées dans le cadre de la vision par ordinateur restent des sujets ouverts à la recherche. Les principales difficultés à surmonter renvoient à la grande variabilité morphologique du visage et aux déformations locales plus ou moins prononcées liées à la richesse des expressions faciales. En outre, les scènes à analyser, de contenus quelconques, sont acquises par une seule caméra fixe ou mobile, en général non calibrée et dans des conditions d'éclairement *a priori* inconnues et surtout non stabilisées.

L'approche la plus classique mettant en œuvre un modèle d'objet de tête pour analyser des séquences vidéos faciales comporte deux étapes de difficulté croissante : 1) une adaptation globale de la pose du modèle, en assimilant le mouvement de la tête à un mouvement rigide, 2) une adaptation locale de la forme du modèle, pour prendre en compte les déformations/mouvements des différentes parties du visage. En adoptant une telle approche, nous proposons une méthode robuste d'estimation de la pose 3D globale de la tête dans des séquences vidéos acquises dans un contexte réaliste, visant à contrôler les conditions suivantes : 1) acquisition avec une seule caméra non calibrée, fixe ou mobile, 2) conditions non stabilisées d'éclairement, 3) mouvements complexes de grande amplitude, 4) déformations locales liées aux expressions faciales, 5) occultations partielles de la tête. La démarche adoptée consiste en une mise en correspondance de primitives 3D du modèle (géométrie, indice de visibilité) avec des primitives 2D extraites des images (mouvement, texture). La mise en correspondance est effectuée par minimisation d'une fonctionnelle relativement aux paramètres 3D de pose via la méthode du simplexe. Afin de garantir la robustesse et la précision du recalage même pour des mouvements de grande amplitude, cette approche est couplée à une interpolation temporelle non rigide, contrôlée par le champ de déplacement et mise en œuvre au sein d'une modélisation ondulatoire par groupe de paquets d'onde. Les performances de la méthode développée ont été étudiées et validées sur des séquences synthétiques, puis testées sur des séquences réelles comportant des variations d'éclairement, des occultations partielles du visage et des mouvements de grande amplitude. Les résultats en démontrent la fiabilité et la précision.

Concernant la capture des mouvements non rigides du visage et des expressions faciales, nous avons développé une méthode de recalage par prototypes déformables pour le suivi des parties du visage les plus expressives dans une communication (bouche, yeux). La démarche adoptée consiste en une modélisation des prototypes déformables associés aux éléments faciaux d'intérêt par des B-splines interpolant les Paramètres MPEG-4 de Définition du Visage (PDV). Modélisant l'élasticité de chaque prototype par un réseau de ressorts reliant les PDV correspondants, l'énergie interne est exprimée en termes de contraintes d'élasticité et de symétrie locale. L'énergie externe des prototypes, imposant leur interaction avec les données, prend en compte plusieurs primitives d'image (contours, texture, topographie), proprement combinées, afin de renforcer la robustesse du recalage. L'initialisation robuste des prototypes d'une image à l'autre est effectuée en couplant une procédure de segmentation automatique de l'iris et de détection de la configuration ouverte/fermée de l'œil à l'estimation de la pose 3D de la tête précédemment développé. La stabilité et la précision des résultats sont établies à partir de séquences vidéos de scènes d'intérieur ou d'extérieur acquises dans des conditions réalistes. Enfin, la procédure de suivi de primitives de visage a été intégrée dans un schéma d'analyse/synthèse de déformations faciales, compatible MPEG-4, pour l'animation d'avatars à partir de séquences vidéos naturelles.

Mots clef : modèle 3D d'objet, séquences vidéos monoscopiques, estimation de la pose 3D, recalage 3D/2D, texture, flot optique, translation et rotation de grande amplitude, occultation, appariement par bloc, interpolation temporelle, modélisation ondulatoire, critère de visibilité, analyse de déformations faciales, description MPEG-4 du visage, prototype déformable, bouche, yeux, B-splines, classification floue non supervisée, méthode du simplexe, synthèse de déformations faciales.