Modélisation et suivi des déformations faciales : applications à la description des expressions du visage dans le contexte de la langue des signes

Hugo Mercier

Université Toulouse III École Doctorale Informatique et Télécommunications

Formation Doctorale Image, Information et Hypermédia

Modélisation et suivi des déformations faciales

Applications à la description des expressions du visage dans le contexte de la langue des signes

THÈSE

présentée et soutenue publiquement le 22 mars 2007

pour l'obtention du

Doctorat de l'Université Paul Sabatier – Toulouse III

spécialité Informatique

par

Hugo Mercier

Composition du jury

Directeur de thèse :	M. Patrice Dalle	Professeur UPS, Toulouse
Rapporteurs :	M. Gérard Bailly	Directeur de Recherche CNRS
	M. Maurice Milgram	Professeur UPMC, Paris
Examinateur :	M. Franck Davoine	Chargé de Recherche CNRS
Invité :	M. Jacques Sangla	Chargé de veille technologique, WebSourd

web SOURD

Institut de Recherche en Informatique de Toulouse — CNRS INPT UPS UT1

Avant-propos

Le présent document constitue le mémoire de la thèse « Modélisation et suivi des déformations faciales - Applications à la description des expressions du visage dans le contexte de la langue des signes », financée dans le cadre d'une Convention Industrielle de Formation par la Recherche numéro 639/2003 passée entre la société WebSourd et l'université Paul Sabatier du 1^{er} mars 2004 au 31 mars 2007.

Résumé

Le visage joue un rôle prépondérant en langue des signes, notamment par le sens porté par ses expressions. Peu d'études existent sur les expressions faciales en langue des signes ; cela est dû au manque d'outil de description. Dans cette thèse, il s'agit de développer des méthodes permettant la description la plus précise et exhaustive possible des différents mouvements faciaux observables au cours d'une séquence vidéo de langue des signes.

Le formalisme des modèles à apparence active (*Active Appearance Models* - AAM) est utilisé ici pour modéliser le visage en termes de déplacements d'un certain nombre de points d'intérêt et en termes de variations de texture. Quand il est associé à une méthode d'optimisation, ce formalisme permet de trouver les coordonnées des points d'intérêt sur un visage. Nous utilisons ici une méthode d'optimisation dite « à composition inverse », qui permet une implémentation efficace et l'obtention de résultats précis.

Dans le contexte de la langue des signes, les rotations hors-plan et les occultations manuelles sont fréquentes. Il est donc nécessaire de développer des méthodes robustes à ces conditions. Il existe pour cela une variante robuste des méthodes d'optimisation d'AAM qui permet de considérer une image d'entrée éventuellement bruitée. Nous avons étendu cette variante de façon à ce que la détection des occultations puisse se faire de manière automatique, en supposant connu le comportement de l'algorithme dans le cas non-occulté. Le résultat de l'algorithme est alors constitué des coordonnées 2D de chacun des points d'intérêt du modèle en chaque image d'une séquence vidéo, associées éventuellement à un score de confiance. Ces données brutes peuvent ensuite être exploitées dans plusieurs applications.

Nous proposons ainsi comme première application de décrire une séquence vidéo expressive en chaque instant par une combinaison de déformations unitaires activées à des intensités différentes. Une autre application originale consiste à traiter une vidéo de manière à empêcher l'identification d'un visage sans perturber la reconnaissance de ses expressions.

Abstract

The face, and particularly the meaning of its expressions, plays an important role in sign languages. A few studies on facial expressions in sign language exist. This is due to the lack of description tools. In this thesis, we develop methods that allow accurate and comprehensive description of the different facial movements observed during a sign language video.

We use here the Active Appearance Model formalism (AAM) in order to model face, in terms of interest point displacements and texture variations. When used with an optimization method, this formalism allows to find interest point coordinates on a face. We use here an optimization method called "inverse compositional", that can be used to obtain accurate results in an efficient manner.

In the sign language context, out-of-plane rotations and hand occlusions occur frequently. Thus, the development of robust methods is needed. It exists, for that purpose, a robust flavor of the AAM optimization methods that allow to consider the input image as being noisy.

We extended it in order to detect occlusions in an automatic manner, with the assumption that the algorithm behavior in the unoccluded case is known.

The algorithm result consists in 2D coordinates of each interest points in each image of a video sequence, eventually linked with a confidence value. These raw results can then be used for different applications.

We thus propose to describe an expressive video sequence, at each frame, as being a linear combination of unitary facial deformations activated with different intensities. Another original application consists in a video processing that prevents the face from being identified, while keeping unchanged the meaning of its expressions.

Remerciements

Je tiens à remercier Patrice Dalle, grâce à qui l'aventure de cette thèse a pu commencer, pour m'avoir proposé un passionnant sujet de DEA, pour m'avoir introduit à la communauté des sourds toulousains et pour m'avoir laissé une grande liberté dans l'exécution des travaux présentés dans ce mémoire.

Merci à MM. Maurice Milgram et Gérard Bailly d'avoir accepté d'être rapporteurs de cette thèse, et de l'avoir relu dans un délais très court. Merci également à Franck Davoine pour avoir accepté d'être examinateur et pour l'ensemble de ses remarques pertinentes.

Je tiens également à remercier WebSourd, l'entreprise en tant que telle et l'ensemble de ses salariés pour m'avoir fait confiance dès le début. Le développement d'une activité de recherche dans le contexte initial de l'entreprise était ... courageux. Ma confrontation au monde de l'entreprenariat coopératif et au monde de la langue des signes, n'a pas toujours été facile, mais a toujours été d'une grande richesse. Merci donc à tous les associés de WebSourd pour leur confiance ainsi que leur patience et leur pédagogie à l'égard de ma langue des signes naissante : Jacques, François, Martine, Pascal, Christine, Bruno, Nicolas, Marylène, Olivier, Audrey, Frédéric, Christina, Janine.

Merci à l'équipe TCI pour son accueil chaleureux, pour l'agréable ambiance qu'elle peut créer et pour tous les moments de joies (et de doutes) passés entre thésards. Pour toutes nos discussions, pour nos pôts et le reste, merci en particulier à Boris, Fred G., Fred C., Sylvie, Benoît, Patrice, Jean-Denis, Alain, Christophe, Pierre ainsi qu'à ceux qui sont passés par là : Laurent, Rodolfo. Merci aussi à Julien pour nos fructueux échanges.

Merci à mes amis et à mes proches pour leur soutien et leur encouragements et pour m'avoir supporté pendant ces quelques années : Brice, Anh Tho, Nico, Steph, mes parents, JR, Arthur et plus particulièrement Betty.

Et enfin, puisqu'un *geek* en reste un, j'aimerais exprimer mon profond respect à Richard Stallman, Donald Knuth, Linus Torvalds, Jimmy Wales, et dans un autre registre Jimmy Page, John Bonham, ou encore Terry Pratchett et Dan Simmons.

Table des matières

Première partie

Introduction et état de l'art

Chapitre 1

Introduction

Nous nous plaçons dans le contexte de l'analyse vidéo des expressions faciales en langue des signes française (LSF). Il s'agit de développer des outils permettant la description de la langue des signes observée par un système vidéo mono-vision. En particulier, nous nous intéressons aux modèles, méthodes et outils logiciels permettant la description la plus exhaustive des mouvements faciaux observés sur une vidéo.

1.1 Langue des signes française

La langue des signes (LS) est la langue naturelle des sourds. C'est une langue visuo-gestuelle : elle utilise le canal visuel en entrée et le canal gestuel en sortie. Le terme gestuel doit être compris au sens large : la production d'un message fait intervenir de manière prépondérante les deux mains, le visage et le haut du corps (buste et épaules).

Bien qu'encore objet de nombreuses recherches, notamment par des linguistes, le fonctionnement de la langue des signes commence à être bien connu. La grammaire des langues des signes est principalement spatiale : un espace virtuel placé devant le signeur (et partagé avec son ou ses interlocuteurs) permet de mettre en relation plusieurs objets linguistiques. Tout au long du discours, cet espace, appelé *espace de signation*, peut être successivement rempli de nouveaux éléments, localement référencé, pertinisé, dé-référencé, ou vidé.

Les éléments lexicaux peuvent être placés dans l'espace de signation suivant une logique temporelle (sur l'axe sagittal pour un repère temporel relatif au temps de l'énonciation et sur l'axe latéral pour un repère temporel absolu), suivant une organisation reflétant la spatialité du signifié ou encore suivant une organisation visant au confort gestuel du signeur.

Ces différents éléments peuvent être liés par une action (l'un actif, l'autre passif), ou bien comme qualificatif l'un de l'autre (équivalent de l'adjectif).

Les modes de fonctionnement de la grammaire spatiale semblent être en grande partie partagés par les différentes langues des signes à travers le monde

(à l'exception de certains éléments culturels comme par exemple le sens des axes temporels). A l'inverse, cette grammaire particulière est difficilement comparable à celle des langues vocales, bien qu'on puisse voir également certains éléments non-verbaux peu formalisés entrer en jeu dans les langues vocales (placement de concept dans l'espace puis référencements ultérieurs).

Les éléments lexicaux de la langue eux, sont beaucoup plus dépendant de la culture et varient donc d'une langue des signes à l'autre ; ils sont composés d'objets appelés communément « signes ». Ils sont définis par l'emplacement, la configuration, l'orientation et le mouvement des mains ainsi que par une expression faciale. Certains signes peuvent être légèrement modifiés par l'exagération de leurs paramètres (vitesse, ampleur) et en particulier par la modification de l'expression faciale.

La description de formes ou d'aspects s'effectue par un signe reflétant de manière générique la forme de l'objet (ou bien de ses contours) et par une expression particulière (ainsi le gonflement des joues correspond généralement à quelque chose d'important, de gros et la succion des joues à quelque chose de fin, maigre).

D'un point de vue grammatical, la plupart des modes du discours (assertatif, dubitatif, capacitif, interrogatif, négatif, etc.) sont définis par une expression faciale. L'expression faciale joue donc plusieurs rôles : composant lexical, élément syntaxique, modalité ou valeur emphatique.

Le regard joue aussi un rôle prépondérant puisqu'il permet la construction logique de la phrase par le référencement à l'espace de signation et le changement de contexte entre l'énonciation « standard » où le regard est porté sur l'interlocuteur et les transferts personnels, où le locuteur prend le rôle d'un personnage, voire d'un objet.

Observé depuis une caméra fixe, le visage du locuteur peut être très fréquemment en rotation parce que le regard est porté sur une zone particulière de l'espace de signation ou bien parce qu'un personnage est joué. Les transferts peuvent amener le haut du corps tout entier à se mouvoir.

L'utilisation de l'espace de signation peut de même entraîner des placements de signes qui occultent en partie le visage du signeur. Le visage peut être occulté partiellement si un signe doit se faire près du visage ou bien encore quand le visage est en rotation hors-plan.

Enfin la langue des signes n'a pas de forme écrite propre, son analyse s'approche de l'analyse des langues orales dans le sens où l'on doit traiter toutes les caractéristiques propres à l'identité de la personne qui signe : la vitesse à laquelle elle signe, l'expressivité du visage et du corps, l'intensité de ses signes, etc. Ceci peut être considéré comme l'équivalent de la prosodie des langues vocales, c'est-à-dire les effets (sonores) annexes de la langue, dont le non-respect ne modifie pas fortement la compréhension d'un énoncé.

1.2 Problématique

La problématique de cette thèse est l'étude des méthodes permettant le développement d'outils logiciels de description des mouvements faciaux observés sur une vidéo de langue des signes. Ceci implique une modélisation des expressions, sa représentation informatique et le développement de méthodes d'extraction des caractéristiques de cette description.

Les mouvements faciaux sont les mouvements musculaires possibles d'un visage, ils comprennent les rotations du crâne.

Il s'agit de fournir automatiquement une description précise et exhaustive des mouvements faciaux observés lors d'une vidéo de langue des signes dans le but :

- d'aider à l'analyse linguistique de séquences vidéos de langue des signes,
- de permettre certains traitements automatiques de l'image du visage (réalité augmentée, modification de l'apparence, anonymisation),
- de reconnaître les expressions dans le but d'une aide à la traduction,

La séquence vidéo représente une production signée d'une seule personne, filmée dans des conditions d'éclairage qui n'évoluent pas au cours du temps.

On suppose que le visage du locuteur est déjà en partie connu du système, ce qui implique qu'un travail (manuel dans notre cas) de calibrage soit effectué au préalable. Le but est que ce travail manuel demandé sur la phase de calibrage du système soit minimal et peu contraignant et que le système soit capable par la suite de suivre avec précision les déformations faciales. La langue des signes posant un contexte difficile d'analyse (présence de rotations hors-plan et d'occultations par les mains), le système doit être robuste. En particulier, en cas d'échec de l'analyse, le système devra être capable de détecter cette erreur, de proposer une stratégie de reprise et d'en informer l'utilisateur.

Nous détaillons dans le chapitre suivant un état de l'art relatif à l'analyse du visage en général et à la description des expressions en particulier, en décrivant les formalismes utilisés pour la description manuelle et informatique ainsi que quelques techniques d'analyse automatique présentes dans la littérature. Nous détaillerons en particulier l'approche basée sur les modèles déformables qui nous semble bien adaptée à notre problématique.

Dans le chapitre 3, nous détaillons la mise en œuvre d'un des algorithmes utilisant des modèles à apparence active : l'algorithme à composition inverse. Celui-ci sera évalué dans le chapitre 4.

Dans le chapitre 5 nous présentons une amélioration d'un des algorithme d'adaptation de modèle à apparence active permettant de prendre en compte les occultations manuelles, qui sont fréquentes en langue des signes. Ce chapitre présente aussi une stratégie de suivi de déformations sur une séquence vidéo.

Enfin, dans le chapitre 6 est présentée la manière dont les résultats de l'algorithme de suivi peuvent être exploités pour la description des expressions. Certaines applications, telles que l'anonymisation, spécifique au contexte de la langue des signes, sont présentées.

Chapitre 2

État de l'art

L'analyse du visage est une discipline en plein essor dans la communauté du traitement d'images et de la vision par ordinateur. Nous définissons ici dans un premier temps les différentes problématiques existantes faisant partie du domaine général de l'analyse du visage en donnant, dans la mesure du possible, les références bibliographiques importantes pour chaque problématique.

Dans un deuxième temps, nous nous focalisons sur la description des expressions en présentant différents formalismes qui ont été utilisés dans la littérature.

Nous présentons différentes méthodes d'analyse du visage et plus spécifiquement des expressions en les classant soit dans les approches dites « globales », sans segmentation soit dans les approches avec segmentation préalable.

Enfin, nous présentons un tour d'horizon des méthodes d'analyse basées sur le formalisme des modèles déformables.

2.1 Tâches de l'analyse du visage

L'analyse du visage est une discipline dont les premiers travaux sont liés à l'essor de l'Intelligence Artificielle, discipline phare de l'informatique des années 1960. En effet, les premiers travaux sur l'analyse automatique du visage remontent aux travaux de Sakai, Nagao et Fujibayashi [Sakai 69] qui proposent un système permettant de détecter l'existence ou l'absence d'un visage dans une image. Suivent les travaux de Kelly [Kelly 70] qui, à partir de trois images de chaque individu (une image du corps, une image de l'arrière-plan et une image du visage), propose une extraction des contours de la tête et une localisation des yeux, du nez et de la bouche. En 1973, Takeo Kanade présente un système de reconnaissance automatique de visage, basé sur une seule image [Kanade 73].

2.1.1 Détection de visage

La détection de visage consiste à déterminer la présence ou l'absence de visages dans une image et en cas de présence à déterminer sa localisation. C'est une tâche préliminaire nécessaire à la plupart des techniques d'analyse du visage. Les techniques utilisées sont généralement issues du domaine de la reconnaissance des formes. En effet, le problème peut être vu comme la détection de caractéristiques communes à l'ensemble des visages humains : il s'agit de comparer une image à un modèle générique de visage et d'indiquer s'il y a ou non ressemblance. Ces méthodes seront donc fortement conditionnées par le choix effectué pour modéliser un visage.

La sortie d'un détecteur de visage indique le nombre de visages présents dans l'image. De plus, la plupart des détecteurs de visage actuels sont aussi des localisateurs de visages : ils renvoient une localisation des visages détectés (une boîte englobante par exemple).

Un état de l'art des méthodes de détection de visage peut être trouvé dans [Yang 02] et [Hjelmas 01]. Il est à noter qu'une technique populaire, qui n'est pas répertoriée dans les articles précédents, a été développée depuis : il s'agit de la méthode présentée dans [Viola 04].

2.1.2 Reconnaissance de visage

La reconnaissance de visage consiste à associer une identité à un visage après l'avoir détecté. On rencontre deux cas différents :

- l'identification où il s'agit de trouver dans une base de données de visages, le visage le plus ressemblant à celui étudié
- l'authentification où il s'agit de vérifier que le visage étudié a bien l'identité qu'il prétend posséder.

Ces deux manières de poser le problème font intervenir des opérateurs d'analyse très différents.

Les systèmes d'identification de visages possèdent une base de données sur laquelle est effectuée un apprentissage. Cette base définit les différentes identités connues du système. Une nouvelle image est présentée au système et le but est de décider à quelle identité connue appartient ce visage ou s'il ne s'agit d'aucun des visages connus.

Les traitements d'un système d'identification de visages peuvent être séparés en deux étages distincts : un premier pour trouver une représentation du visage qui permette de regrouper les visages de la même identité et de discriminer les différentes identités (modélisation) et un étage pour trouver la classe la plus vraisemblable, lors de la présentation d'un nouveau visage (reconnaissance).

Les principales difficultés d'un système de reconnaissance de visage sont la robustesse aux changements d'expressions, de pose, d'illumination, ainsi

qu'aux changements morphologiques dus à l'âge et ceux dus à la présence d'artefacts visuels comme des lunettes ou la barbe.

L'extraction des caractéristiques peut consister en un ensemble de mesures discriminantes de l'identité : le visage est segmenté en composantes (nez, bouche, yeux, etc.) et certaines propriétés locales sont extraites. Une approche plus récente considère le visage dans son ensemble ce qui évite la prise de décision sur la segmentation du visage en composantes à considérer ; la décision est reportée à l'étape de classification.

Un état de l'art des techniques de reconnaissance de visage peut être trouvé dans [Zhao 00] et [Gross 01].

2.1.3 Reconnaissance d'émotions

La reconnaissance d'émotions consiste à associer une émotion à une image de visage. Le but est donc de déterminer, d'après son visage, l'état émotionnel interne de la personne. L'ensemble considéré des émotions affichables par un visage est généralement de petite taille : il s'agit de l'ensemble des sept émotions universelles présenté par Ekman (voir en 2.2.1).

Il s'agit d'un problème du même ordre que la reconnaissance de visage : le visage en entrée doit être classé parmi un ensemble fini de classes représentant les émotions. Cependant, ici les caractéristiques extraites doivent être indépendantes de l'identité (et de la pose, illumination, etc.)

Les techniques utilisées sont donc très proches de celles utilisées pour la reconnaissance de visage, seules vont être changées les composantes faciales à prendre en compte pour la représentation d'un visage.

2.1.4 Description des mouvements faciaux

La description des mouvements faciaux consiste à fournir une description la plus fine et précise possible des différents mouvements musculaires faciaux observés sur une image ou au cours d'une séquence vidéo. Aucune interprétation du sens n'est associée aux mouvements faciaux extraits : il s'agit d'une description de bas niveau, préalable à d'autres études (analyse linguistique, synthèse de visages, etc.)

Le but de la plupart des travaux affrontant cette problématique est de remplacer la fastidieuse analyse manuelle des mouvements faciaux qui ne peut se faire que par des experts analysant longuement chaque image. Ainsi, le but est de fournir une description selon un formalisme jusqu'ici utilisé par les experts de la description des mouvements faciaux : FACS (voir en 2.2.2).

C'est un système de codification des mouvements musculaires du visage qui permet de renseigner sur l'activation et le degré d'activation d'un muscle du visage uniquement à partir d'observations visuelles du visage. Un tel système n'a pas été développé pour une analyse informatique. En particulier, certaines observations sont décrites de manière subjective. Cependant, il s'est par la

suite imposé comme un standard *de facto* de description des expressions et ce même pour les informaticiens.

Les principales difficultés rencontrées sont le besoin d'une description qui peut être très complexe suivant l'application, ainsi que la robustesse aux changements d'identité, de pose et les différentes occultations qui peuvent intervenir.

Le travail présenté dans ce mémoire se situe dans cette problématique de description automatique des mouvements faciaux. La section suivante traite plus en détail les différents formalismes de description utilisés dans la littérature.

2.2 Formalismes de description des expressions

Nous présentons dans cette section les différents formalismes de description des mouvements faciaux utilisés soit pour l'analyse psychologique, soit pour l'analyse (et en particulier la transcription) des langues des signes. Nous présentons de plus la manière dont les mouvements faciaux peuvent être représentés par ordinateur.

2.2.1 Historique

Au XIX^{e} siècle, Guillaume Duchenne de Boulogne est le premier à localiser individuellement les différents muscles faciaux par activation électrique. Des électrodes permettant l'activation des muscles ont été implantées sur une personne souffrant de paralysie faciale. Il est un des premiers à livrer à la communauté scientifique un ensemble de photographies montrant l'activation des différents muscles faciaux. Il montre ainsi que le sourire sincère reflétant une émotion positive est effectué via l'activation de muscles péri-oculaires à la différence du sourire « posé » qui n'affecte que les muscles de la bouche. Ce type de sourire est appelé « sourire de Duchenne » en son honneur. Le muscle *orbicularis occuli* activé lors d'un sourire sincère ne peut, pour la plupart des personnes, pas être activé volontairement.

En 1872, Charles Darwin, dans le cadre de sa théorie sur l'origine des espèces, publie *The Expression of the Emotions in Man and Animals* (voir [Darwin 01] pour une traduction française) dans lequel il émet la théorie de l'universalité des émotions chez l'homme et les animaux. Pour lui les « états d'esprit » sont reflétés de la même façon chez tous les hommes et chez les animaux. Les expressions du visage sont un des vecteurs des émotions.

Ces travaux sont repris au XX^{e} siècle, notamment par l'anthropologue Ekman [Ekman 69] qui vérifie que l'expression de certaines émotions est universellement reconnue à travers le monde. Les émotions choisies étaient la joie, la peur, la surprise, la colère, la tristesse et le dégoût.

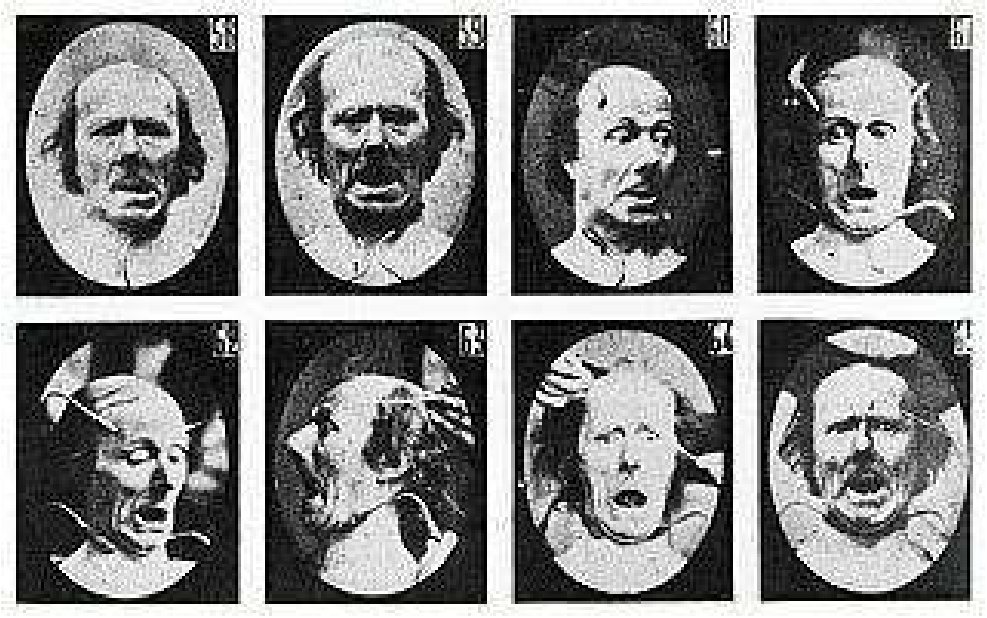


FIG. 2.1 – Photographies des expériences d'activation électrique des muscles de Duchenne

2.2.2 FACS

Le système FACS (*Facial Action Coding System*) développé par Ekman et Friesen [Ekman 78] est un système de description exhaustif des mouvements faciaux. Il s'agit d'associer un code à chaque activation musculaire du visage qui peut être distinguée visuellement. Ces éléments atomiques sont appelés *Action Units* ou AU. Le manuel du codeur FACS contient ainsi la description visuelle des changements du visage lors de l'occurrence de chaque AU ou chaque combinaison d'AUs. De plus, chaque AU peut être affichée avec une amplitude différente. Les auteurs ont retenu un maximum de 5 amplitudes pour chaque AU.

Une *Action Unit* ne correspond pas nécessairement à un muscle facial isolé. En effet, la structure musculaire du visage fait que l'activation d'un muscle entraîne le déplacement de ses voisins dans bien des cas (ceci parce que les muscles sont attachés davantage entre eux qu'aux muscles du crâne).

Une longue étude est nécessaire pour obtenir un niveau d'expertise nécessaire au codage des mouvements faciaux. De plus, même pour un expert, l'analyse requiert un travail très long et fastidieux.

On distingue 46 AUs, pour la description des expressions faciales humaines. Cependant, Scherer *et al.* [Scherer 82] ont observé environ 7000 combinaisons différentes d'*action units* lors de la production d'expressions spontanées.

De plus, la combinaison de plusieurs mouvements faciaux ne peut pas être décrite facilement par la combinaison visuelle de chacun des mouvements iso-

FIG. 2.2 – Exemples d'*action units*

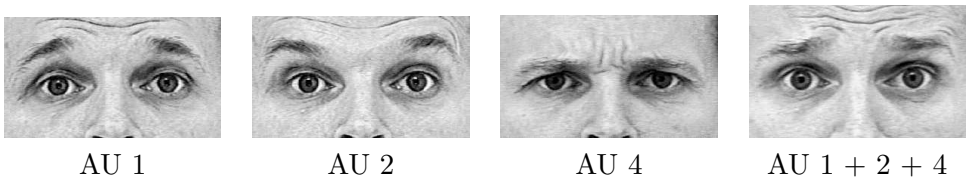


FIG. 2.3 – Exemple de coarticulation des *action units*

lés. En effet, il se produit un phénomène de coarticulation, de la même manière que la prononciation d'un mot ne peut pas être réduite à la concaténation de la prononciation de chacun de ses phonèmes (voir figure 2.3).

2.2.3 En langue des signes

Les recherches autour de la langue des signes suivent la même évolution que celles des langues vocales : la partie non gestuelle (comme la partie non verbale des langues orales) de la langue n'est dans un premier temps pas étudiée, car jugée sans intérêt [Régent 04].

Au XIX^{e} siècle, Bébian [Bébian 25] s'intéresse à une écriture de la langue des signes à visée notamment pédagogique. Il utilise un alphabet d'environ 200 symboles pour écrire la langue des signes, comprenant l'emplacement, la configuration, le mouvement du signe et les expressions faciales, qu'il nomme alors « points physionomiques ». Dans ce système de notation, 11 symboles servent à décrire les expressions faciales (de manière globale et non comme la combinaison de mouvements faciaux) ; ils sont dédoublés (à chaque symbole à valeur « positive » est associé un symbole à valeur « négative ») et échelonnés selon trois intensités. C'est le système le plus riche d'analyse de la langue des signes qui sera publié jusqu'à très récemment.

Dans les années 1960, Stokoe, linguiste, se propose de décomposer la langue gestuelle comme une combinaison de *chérèmes* (par analogie aux phonèmes des langues vocales) qui sont au nombre de 55. Les expressions faciales ne sont pas

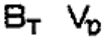
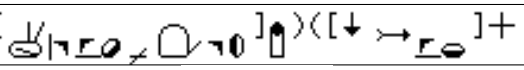
Image	
Stokoe	
HamNoSys	
SignWriting	

FIG. 2.4 – Notations du signe [LIRE] de la langue des signes américaine dans les différents systèmes : Stokoe, Hamnosys et SignWriting

considérées.

Dix ans plus tard, l'université de Hambourg propose un système de notation permettant une analyse détaillée de la langue des signes. Ce système (appelé HamNoSys) est donc dédié dans un premier temps à une communication entre chercheurs linguistes. HamNosys est constitué d'environ 500 symboles qui reprennent le principe des chérèmes de Stokoe. Le système évolue fréquemment et les expressions n'y apparaissent que très tardivement. A l'heure actuelle, la notation des expressions faciales est en cours de développement.

Dans les années 1980, un linguiste français, Christian Cuxac, s'intéresse à la langue des signes, et en particulier à ce qui reste appelé la pantomime. Il développe alors sa théorie de double visée : un locuteur confirmé en langue des signes jongle sans cesse entre deux modes d'expression : *dire en montrant* et *dire sans montrer*. Le premier (appelé aussi visée illustrative) est composé de structures dites de grande iconicité, où la production gestuelle a un rapport direct avec la réalité (ce qui était appelé pantomime). Le deuxième (hors visée illustrative) n'a pas de rapport direct avec la réalité et est composé de signes codifiés, dits standards.

Cuxac insiste sur le rôle des expressions faciales en langue des signes et il isole en particulier :

- une valeur aspectuelle de l'expression, permettant de préciser les différents aspects d'une description (taille, poids, etc.),
- une valeur modale, permettant d'introduire les modes du discours (interrogatif, assertatif, conditionnel, etc.).

Il a étudié la langue des signes à travers une méthode d'annotation de vidéos. L'analyse linguistique est faite via une représentation « en partition » où

chaque ligne représente plusieurs composantes du signe ou du discours. En particulier, les expressions sont annotées. Cependant, les outils de description ont manqué et il est généralement uniquement reporté la présence d'une expression faciale, par la notation « MF » (pour mimique faciale) sans rien préciser d'autre, excepté sa durée. Dans quelques cas, les expressions sont annotées avec une description informelle en français, sans suivre de norme particulière. Il en résulte des descriptions difficiles à interpréter telles que « moue dubitative » par exemple.

Parallèlement à ces différentes recherches linguistiques, un système d'écriture des langues des signes, nommé SignWriting a été développé comme extension d'un système de notation de chorégraphie, DanceWriting. C'est un système qui s'est construit avec peu de rapport avec la communauté de la recherche linguistique et pourtant le seul à avoir été et à être utilisé par certaines communautés de sourds. C'est un système proche du dessin. De nombreux symboles permettent d'annoter la configuration des mains, l'emplacement et le mouvement.

SignWriting est très riche en ce qui concerne la description des expressions faciales : 105 symboles existent pour leurs descriptions. Cependant, les différents symboles des expressions ont été créés sans considérations linguistiques ou physiologiques : ainsi il s'agit de la combinaison des parties du visage et des variantes possibles pour chaque partie. Dans cette liste de symboles, certains sont alors impossible à réaliser physiquement (« regard arrière » par exemple).

2.2.4 Bilan

Après analyse, il semble que tous les mouvements faciaux aient une importance en langue des signes. Ainsi, l'idéal pour la description automatique d'expressions est un système permettant l'analyse selon une liste exhaustive des mouvements faciaux réalisables, et dans ce cadre FACS s'avère être un formalisme adapté.

Les études existantes sur la description systématique des expressions faciales ont été menées généralement dans le contexte de l'analyse de la gestuelle co-verbale et non de la langue des signes. Il semblerait que la plupart des mouvements faciaux étudiés dans le co-verbal soient importants en langue de signes. En revanche, certains mouvements faciaux sont bien plus fréquents et chargés de sens en langue des signes que dans le co-verbal et donc généralement peu étudiés.

Il s'agit en particulier des mouvements faciaux utilisés pour les descriptions de forme :

- le gonflement des joues (AU34) pour la description de quelque chose de gros, lourd ;
- la succion des joues (AU35) pour la description de quelque chose de fin, maigre ;

– le pincement des lèvres (AU28) pour la description de quelque chose de plat, vide, nu ;
– le plissement des yeux (AU7) pour la description de quelque chose de lointain, flou ;
– les joues gonflées par la langue (AU36) pour la description de quelque chose de caché, discret, interdit ;
– la langue entre les dents (AU37) accompagné d'un souffle d'air pour la description de quelque chose de flasque.

2.2.5 Modèles informatiques

La norme de codage vidéo MPEG-4 [MPEG Working Group on Visual 01] dispose d'un modèle du visage humain développé par le groupe d'intérêt « Face and Body AdHoc Group ». C'est un modèle déformable, capable de s'adapter à une morphologie particulière et à des expressions particulières (voir Fig. 2.5).

Ce modèle est construit sur un ensemble d'attributs faciaux, appelés « Facial Feature Points » (FFP). Des mesures sur ces FFP sont effectuées pour former des unités de mesure (Facial Animation Parameter Units) qui servent à la description des mouvements musculaires (Facial Animation Parameters - équivalents des Actions Units d'Ekman).

Les *Facial Animation Parameter Units* (FAPU) permettent de définir des mouvements élémentaires du visage de manière transposable. En effet, il est difficile de définir les mouvements élémentaires des muscles de manière absolue, mais on peut considérer l'intensité de leur déplacement relative à certaines distances pertinentes comme constant. C'est ce qui permet de donner des expressions humaines à des personnages non-humains.

Comme exemples de FAPU, on peut citer *la largeur de la bouche*, *la distance de séparation entre la bouche et le nez*, *la distance de séparation entre les yeux et le nez*, etc.

Par exemple, l'étirement du coin de la lèvre gauche (Facial Animation Parameter 6 `stretch_l_cornerlip`) est défini comme le déplacement vers la droite du coin de la lèvre gauche d'une distance égale à la longueur de la bouche. Les FAPUs sont donc des mesures qui permettent de décrire des mouvements élémentaires et donc des animations.

Cependant, le niveau de granularité des *Facial Animation Parameters* (FAP) de MPEG-4 est très bas : il s'agit des déplacements du modèle de visage les plus élémentaires. Un mouvement réaliste est ainsi généralement composé de plusieurs *FAP*. Par exemple, l'AU 26 de FACS (« Jaw Drop ») décrit le mouvement d'abaissement du menton ; cet abaissement est accompagné d'un abaissement de la lèvre inférieure. L'abaissement du menton de MPEG-4 (FAP 3 - `open_jaw`) ne décrit pas l'abaissement de la lèvre inférieure : la description n'est donc pas réaliste d'un point de vue musculaire.

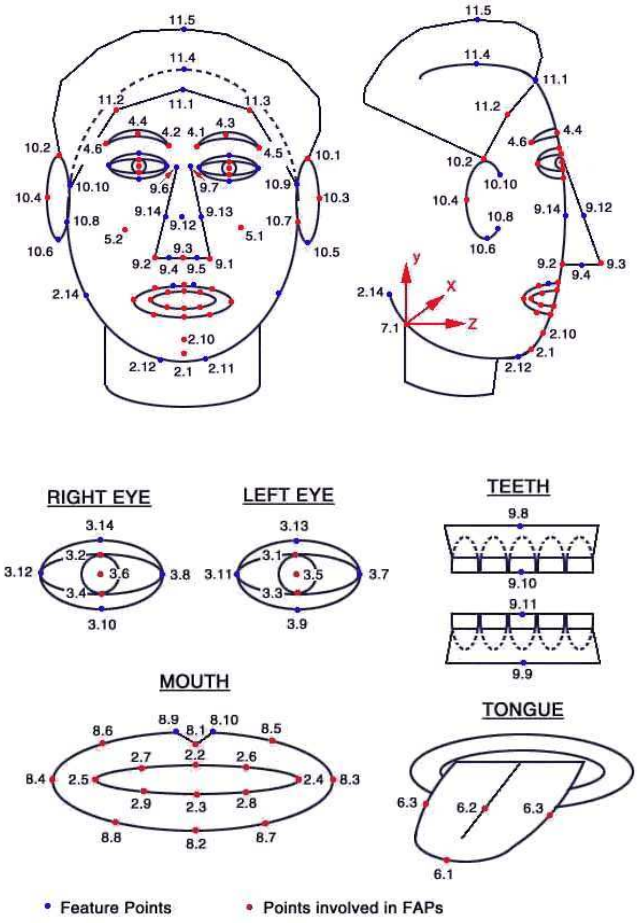


FIG. 2.5 – Modèle de visage MPEG-4 – Facial Animation Parameter Points et Facial Animation Parameter Units

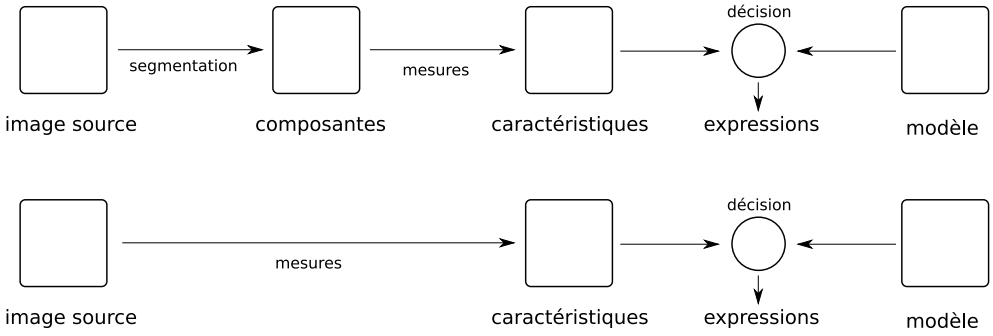


FIG. 2.6 – Principe de l'analyse par segmentation en composantes faciales (en haut) et globale (en bas).

2.3 Méthodes d'analyse du visage

Nous présentons dans cette section les différentes modélisations informatiques du visage utilisées dans la littérature. Il peut s'agir de modèles qui servent aussi bien à l'analyse de l'identité qu'à l'analyse des expressions.

Nous avons distingué deux types de méthodes : celles dites « globales », considérant le visage dans son ensemble sans traitement particulier pour certaines composantes et celles basées sur une segmentation explicite du visage en composantes et une description des caractéristiques de ces composantes faciales (voir Fig. 2.6).

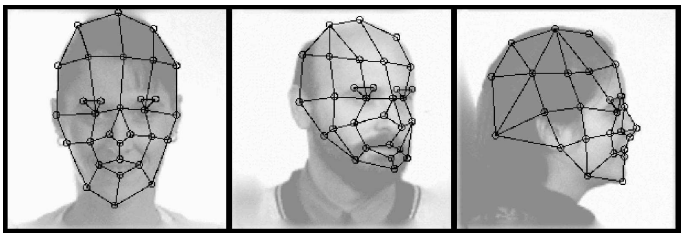
De plus, nous présentons en tant que méthodes hybrides, les différentes méthodes basées sur l'utilisation de modèles déformables (à forme active ou à apparence active). Ces méthodes sont présentées en détail dans un cadre commun d'analyse.

Il est à noter qu'un état de l'art sur les techniques d'analyse automatique des expressions faciales est disponible dans [Pantic 00] et [Fasel 03].

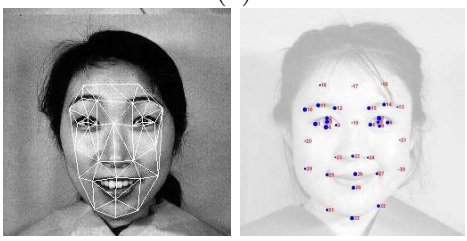
2.3.1 Approche globale sans segmentation

La modélisation la plus simple du visage consiste à prendre en compte un ensemble de points du visage représentant l'état de certaines composantes. Ces points doivent correspondre à des indices visuels qu'il est possible de mettre en correspondance sur toutes les observations de l'étude. Les points à analyser sont différents quand il s'agit d'analyser l'identité de quand il s'agit d'analyser l'expression.

Un visage peut être caractérisé par les coordonnées de chacun des points du modèle ainsi que par la valeur des pixels en leur voisinage, permettant de définir un descripteur de visage plus puissant que l'image brute. Certaines méthodes considèrent un traitement particulier en chacun des points d'intérêt : le résultat du traitement en chacun des points formant le vecteur d'entrée du système d'analyse.



(a)



(b) (c)

FIG. 2.7 – Modèles de visage basé sur un maillage de points. (a) modèle pour la reconnaissance d'identité [Wiskott 97]. (b) modèle pour la reconnaissance d'émotions [Lyons 98]. (c) importance des points pour la reconnaissance d'émotions [Zhang 98].

Par exemple, la transformée par ondelettes de Gabor[1]peut être utilisée sur une grille de points aussi bien pour la reconnaissance d'identité [Wiskott 97] que pour la reconnaissance d'émotions [Lyons 98]. L'ensemble des réponses des filtres de Gabor forme un vecteur transmis en entrée à un système de reconnaissance. Dans [Zhang 98], ces vecteurs d'entrée sont présentés à un réseau de neurones pour la reconnaissance d'émotions. Après un ensemble d'expériences sur les points à choisir et sur les paramètres du réseau de neurones, l'auteur conclut sur l'importance de chacun des points choisis : les points les plus importants pour la tâche de reconnaissance des émotions sont les points autour des yeux, de la bouche, des sourcils et du menton.

Littlewort Littlewort *et al.* [Littlewort 06] proposent un système d'analyse automatique des expressions faciales. Il s'agit d'un système de classification permettant de détecter la présence d'*action units* ainsi que leur intensité au cours d'une vidéo. L'extraction des données utiles à la classification se fait à partir de la texture du visage quasiment brute : le visage et les yeux sont détectés par des techniques proches de celles développées par Viola et Jones

[1]Pour rappel, le filtrage de Gabor consiste à convoluer l'image par une fonction qui est le produit d'une gaussienne et d'une sinusoïdale. Elle possède plusieurs paramètres : la largeur (variance de la gaussienne), la fréquence et l'orientation (paramètres de la fonction sinusoïdale). L'utilisation de ce filtre est justifiée par le fait qu'un traitement semblable existe dans le cortex visuel primaire. De plus, les réponses d'un ensemble de filtres donnent une signature relativement discriminante utilisée pour la reconnaissance d'objets.

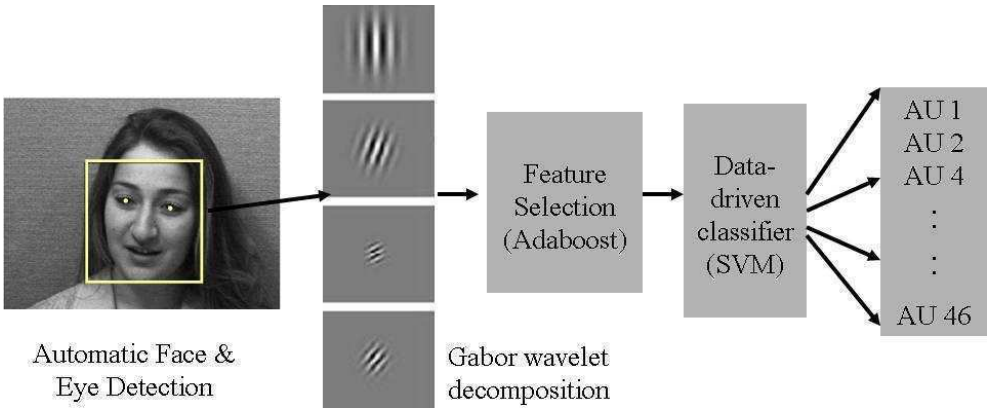


FIG. 2.8 – Vue schématique du système de reconnaissance de [Littlewort 06]

dans [Viola 04]. L'image du visage est alors normalisée en une fenêtre de 96×96 pixels où les yeux sont à une position fixe. Les caractéristiques extraites sont les réponses d'un ensemble de filtres de Gabor à différentes échelles et orientations en chacun des pixels de l'image (représentant au total plus de 650 000 filtres).

Les réponses des filtres de Gabor sont alors données en entrée à des SVM[2] (un SVM est utilisé pour chaque *action unit* à détecter). Chaque SVM a été précédemment entraîné sur la base Cohn-Kanade [Kanade 00] : la présence de l'*action unit* est la réponse positive, et une réponse négative est renvoyée dans tous les autres cas.

Le système dans son ensemble permet un taux de reconnaissance de plus de 90%. De plus, l'utilisation des SVM permet de mesurer l'intensité de l'*action unit* reconnue.

Le principal avantage réside dans le fait que les techniques d'apprentissage et de classification sont génériques et il est donc aisé d'ajouter de nouvelles *action units* à détecter, à condition d'avoir la base d'apprentissage correspondante. Cependant, le système n'est capable que de traiter des visage vus de face et sans occultations.

2.3.2 Approche avec segmentation en composantes

L'approche basée sur l'analyse des composantes faciales consiste à employer une méthode particulière d'analyse pour chacune des composantes faciales.

[2] *Support Vector Machine* (Machine à Vecteurs de Support ou Séparateur à Vaste Marge) : méthode de classification linéaire qui permet de trouver un hyperplan séparateur qui maximise la marge entre les différentes classes tout en minimisant l'erreur de classification. Cette méthode est utilisée généralement pour son extensibilité à la classification non-linéaire en appliquant le « truc du noyau » (*kernel trick*). Une fonction noyau transforme les observations initiales, non-linéairement séparables, en observations linéairement séparables dans un nouvel espace de dimension supérieure. Le *kernel trick* permet alors de travailler dans l'espace transformé sans avoir à calculer explicitement l'image de chaque observation.

Dans cette catégorie, on trouve des techniques basées sur des modèles paramétriques : il s'agit d'un ensemble de points d'intérêt des composantes faciales liés entres eux par certaines contraintes. Il s'agit généralement de contraintes imposées sur la forme de la composante. Par exemple, les lèvres, qui sont des composantes très étudiées, ont des contours bien marqués et peuvent donc être modélisées par des polynômes ou bien encore par des formes plus libres. Les techniques d'analyse consistent alors à superposer les contours du modèle utilisé avec les contours réels ; il s'agit généralement de méthode d'optimisation, maximisant un critère de ressemblance (la répartition des points à fort contraste sur les contours du modèle par exemple).

La segmentation nécessite généralement plusieurs étapes. Dans une première étape, certains points caractéristiques de la composante sont localisés. La localisation s'effectue par analyse des propriétés bas niveau de l'image : analyse du contour, du contraste (après transformation éventuelle de l'espace des couleurs RGB), etc. Une deuxième étape consiste à faire passer une courbe paramétrique par ces premiers points et à considérer que le contour de la composante est décrit par le segment de courbe paramétrique. D'autres allers-retours entre l'image et le modèle peuvent éventuellement être ajoutés pour rendre plus robuste la segmentation.

On trouve ainsi dans [Eveno 03] ou [Hammal 06] une telle approche. Dans [Eveno 03], l'auteur présente une méthode de segmentation des lèvres. Les traitements sont effectués sur une grandeur appelée « pseudo-teinte » qui est plus importante pour les lèvres que pour la peau. Le calcul d'un gradient hybride utilisant l'information de pseudo-teinte et de luminance permet de faire ressortir la partie supérieure des lèvres. Une technique de contour actif (dont le déroulement a été modifié pour assurer une meilleure robustesse à l'initialisation), appelée « jumping snake », utilise ces informations pour la localisation de six points caractéristiques de la bouche. Enfin, ces points permettent d'initialiser un ensemble de courbes polynomiales cubiques qui décrivent les contours des lèvres.

Dans [Hammal 06] la technique de segmentation des lèvres de Eveno est reprise. L'auteur y ajoute des méthodes de segmentation des yeux (iris et contours) et des sourcils. La première étape consiste à filtrer l'image fournie en entrée au système de manière à renforcer les gradients tout en étant robuste aux différents éclairages. Les boîtes englobantes des yeux sont détectées en appliquant des contraintes morphologiques. Dans chacune des boîtes, le pixel dont le gradient est maximal est considéré comme faisant partie du contour de l'iris. Le contour de l'iris est détecté en cherchant le centre d'un cercle de rayon fixé. Pour les sourcils, une première estimation des points de contour est obtenu par une analyse des profils horizontaux et verticaux. Des courbes de Bézier sont ensuite adaptées en partant de cette première estimation et en maximisant l'intensité du gradient le long des courbes.

Ces méthodes sont basées sur de nombreuses heuristiques (dans le choix des modèles ou des données de bas niveau extraites de l'image). L'avantage est qu'il n'est pas nécessaire de disposer d'une base d'apprentissage et que les

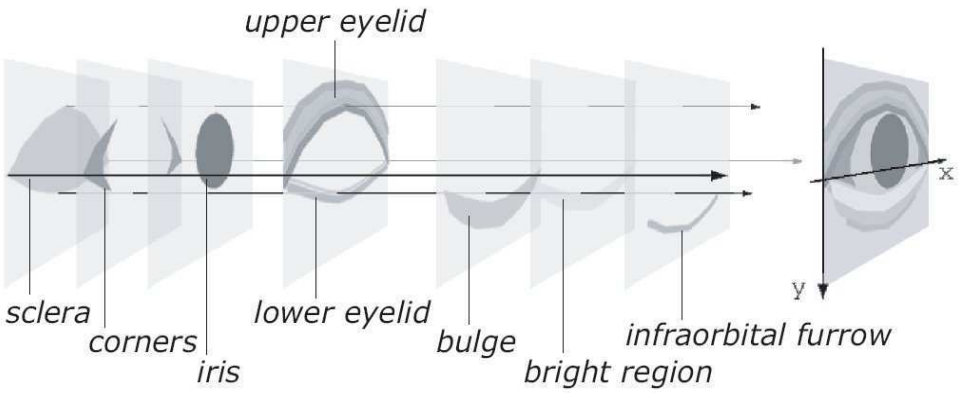


FIG. 2.9 – Modèle génératif de l'œil utilisé dans [Moriyama 06]

.

traitements peuvent donc être potentiellement très robustes. En revanche, il est difficile de savoir si les heuristiques choisies sont les meilleures.

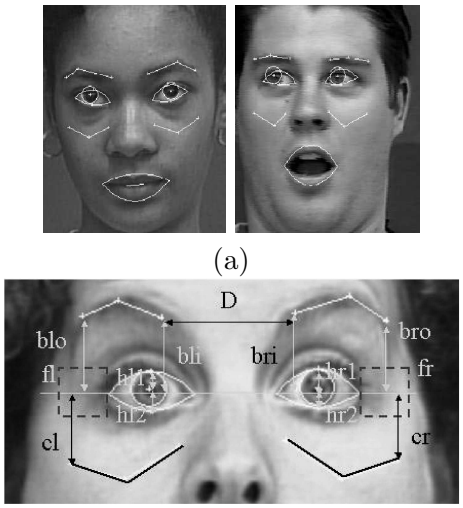
Moriyama Moriyama *et al.* [Moriyama 06] présentent une méthode précise d'analyse des mouvements de l'œil. La méthode est basée sur l'utilisation d'un modèle génératif de l'œil humain : il s'agit d'un modèle 2D texturé, organisé en couches plus ou moins transparentes. L'œil est découpé en plusieurs sous-composantes : paupières, sourcils, iris, etc. auxquelles est associé un ensemble de paramètres de forme et d'aspect (intensité lumineuse et couleur). Les paramètres permettent de faire évoluer le modèle selon une morphologie particulière (paramètres de structure) ou selon une expression particulière (paramètres de mouvement).

Le but de l'algorithme est de trouver les paramètres du modèle à chaque image, afin que celui-ci ressemble le plus possible à l'image observée.

Une fois le visage normalisé en forme (en prenant en compte les rotations et en redressant l'image via l'utilisation d'une méthode de suivi de demi-cylindre 3D) et en luminosité, les auteurs utilisent l'algorithme de Lucas–Kanade, modifié de façon à prendre en compte les déformations possibles du modèle.

Bien que la position du modèle soit initialisée manuellement sur la première image de chaque séquence à analyser, les résultats sont excellents en terme de robustesse et de précision.

Cependant, ce type de modèle est très difficile à développer. En effet la forme de chacune des composantes doit être fidèlement modélisée ainsi que ses variations possibles d'aspect. De plus, les variations de forme et d'aspect doivent être différenciées selon qu'elles sont dues à des variations inter-personnelles (variations morphologiques) ou à des variations intra-personnelles (variations expressives).



(a)

(b)

FIG. 2.10 – Modèle de visage basé sur des courbes paramétrées [li Tian 01]. (a) Modèle. (b) Mesures associées au modèle

Tian Tian *et al.* [li Tian 01] proposent un système d'analyse automatique des expressions faciales, par reconnaissance d'*action units*. La classification est effectuée par des réseaux de neurones. Les données d'entrée des réseaux de neurones sont un ensemble de mesures effectuées sur des descripteurs principalement géométriques du visage, qui sont extraits par des méthodes *ad hoc*. Les sourcils et le haut des joues sont par exemple modélisés par deux segments de droite, la bouche et les yeux par des courbes paramétriques. Les modèles de la bouche et des yeux ont plusieurs états possibles : ouvert, semi-ouvert et fermé.

En plus de ces descripteurs, la présence de rides est détectée par une analyse de contour dans certaines zones (haut du nez par exemple) : l'opérateur de Canny appliqué à ces zones permet de déterminer s'il y a présence de ride en comparant le nombre de contours aux contours présents sur la première image de la séquence.

L'ensemble de ces données est fourni en entrée à des réseaux de neurones multi-couches ayant une sortie par *action unit*. Le système est capable de détecter l'activation de 15 *action units* et certaines combinaisons avec un taux d'environ 90% et un taux de fausses alarmes d'environ 10%.

Bien que le système offre de bonnes performances, les méthodes d'extraction des paramètres sont très spécifiques et construites empiriquement sur des indices de couleur, contours et mouvements. De plus, le système doit être initialisé manuellement sur la première image. Aucune information concernant la robustesse du système aux occultations manuelles et aux rotations du crâne n'est disponible.

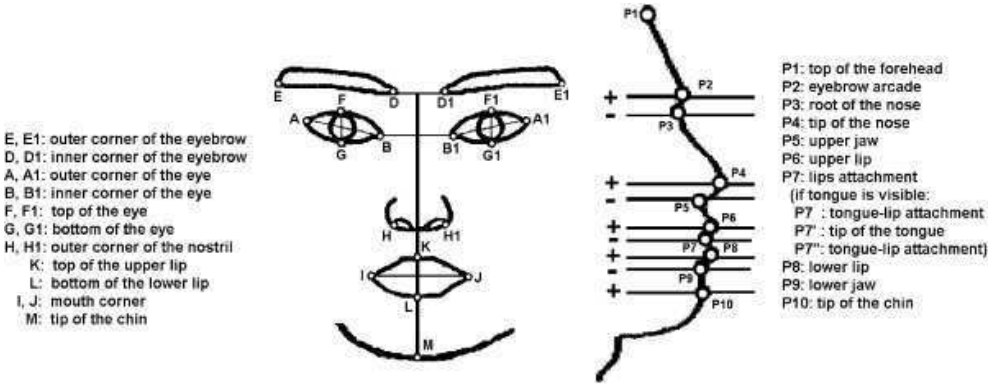


FIG. 2.11 – Ensemble des points d'intérêts du visage utilisé dans [Pantic 04].

Pantic Pantic *et al.* [Pantic 04] proposent un système de reconnaissance d'*action units* à partir d'images statiques de visages de face et de profil.

L'analyse est effectuée dans une première partie par une détection du contour du visage sur l'image de profil. Un ensemble de points d'intérêt est détecté aux zéros de la dérivée du profil. Ils permettent d'initialiser la recherche des points d'intérêt sur l'image de face. Plusieurs détecteurs « classiques » sont appliqués sur l'image de face à partir de cette première initialisation. L'idée est ici de lancer plusieurs détecteurs et de retenir le meilleur, supposant que quand un détecteur *ad hoc* est performant, les autres ne le sont pas. Les points d'intérêt à détecter sont des points « stables », qui ne varient pas avec l'expression (coins des yeux par exemple) et des points en mouvement avec l'expression. La validité de chaque détecteur est déterminée comme étant la distance entre la position détectée des points stables comparée à leur position sur une image de référence (contenant l'expression neutre de la personne). Il est alors supposé que les expressions surviennent en déplaçant le moins possible les points stables du visage par rapport à une image de référence.

Les positions des points d'intérêt (de profil et de face) servent au mécanisme de reconnaissance des *action units*. La reconnaissance est basée sur des règles heuristiques. Le système est capable de reconnaître 32 AUs avec un taux de reconnaissance de 86 %. Cependant, le contexte d'acquisition (images de face et de profil) rend les applications restreintes et le système est très difficile à étendre. Il ne détecte pas, par exemple le gonflement des joues (AU34), nécessaire à la description des expressions en LSF.

2.3.3 Modèles déformables

Nous présentons dans cette section le formalisme des modèles déformables (à forme active et à apparence active) et les différentes méthodes les utilisant.

Modèles à forme active

Les modèles à forme active (*active shape model - ASM*) ont été introduits par Cootes et Taylor [Cootes 92]. Il s'agit de modèles déformables vus comme une extension « intelligente » des *snakes* [Kass 87]. En effet, les formes possibles sont fonction d'un ensemble d'apprentissage, alors que la forme peut être arbitraire pour un *snake*.

C'est une méthode itérative qui permet de faire évoluer le modèle vers sa solution. A partir d'une première estimation des paramètres de pose du modèle (translation, facteur d'échelle et rotation), une recherche de gradient est effectuée sur un segment normal au contour en chaque point.

Le gradient image donne les directions probables de déplacement en chacun des points du modèle. Les paramètres de pose sont modifiés de manière à ce que l'ensemble du modèle se rapproche des déplacements candidats. Puis l'erreur résiduelle sert à modifier chaque point de manière indépendante. Cependant, au lieu d'autoriser un déplacement arbitraire, les déplacements sont reprojetés dans l'espace des formes pour n'autoriser que des déplacements possibles du point de vue statistique.

Une deuxième version des modèles à forme active par la même équipe [Cootes 94], prend en compte une statistique de texture locale en chacun des points du modèle pour la recherche du meilleur déplacement. Plutôt que de considérer uniquement le gradient sur la normale en chacun des points (ce qui implique que les points d'intérêt du modèle sont placés sur des pixels à forts gradients), une statistique des niveaux de gris sur la normale en chacun des points est apprise précédemment.

Les profils de longueur k sur la normale en chacun des points sont extraits sur chaque image de la base d'apprentissage et normalisés en intensité lumineuse. Puis la moyenne et la covariance sont calculées pour le profil en chacun des points. Ainsi, il est possible de calculer une distance de Mahalanobis entre un profil extrait et ceux appris, permettant de juger de la qualité du profil. Lors de la recherche de déplacement sur une nouvelle image, la normale est calculée en chacun des points par un segment de longueur $m \geq k$, puis tous les $(m - k)$ profils possibles sont évalués en calculant leur distance de Mahalanobis. Le point ayant la plus faible distance est choisi comme déplacement candidat et on effectue la procédure de calcul des paramètres de pose et de reprojection dans l'espace des formes comme précédemment.

La statistique de forme est obtenue par une décomposition en composantes principales (voir l'analyse de forme plus loin).

Un point fort des modèles à forme active est qu'ils peuvent prendre en considération une zone variable de l'espace de recherche — en augmentant la longueur des segments servant à la recherche des déplacements candidats, ce qui permet moyennant un coût de calcul important d'être robuste lorsque l'initialisation est mauvaise.

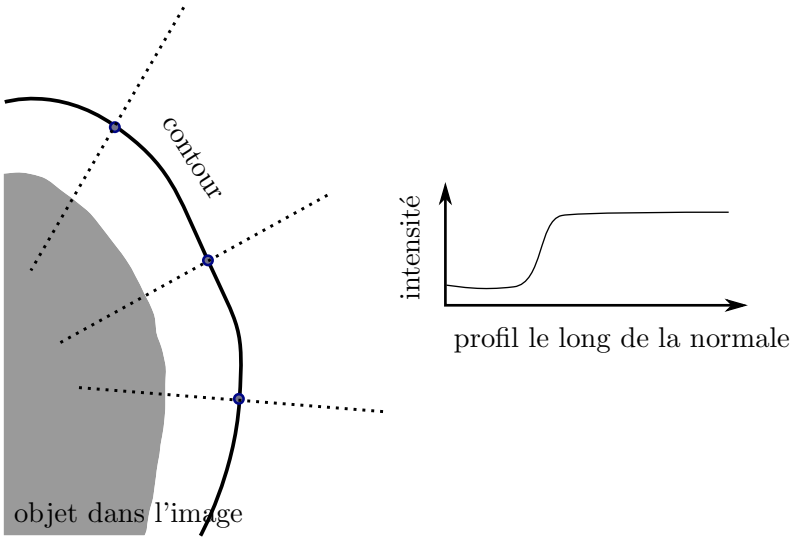


FIG. 2.12 – Schéma d'un modèle à forme active

Modèles à apparence active

Les modèles à apparence active sont une extension des modèles à forme active [Cootes 01]. Les formes possibles ainsi que la texture interne des modèles sont représentées statistiquement à l'aide d'un ensemble d'apprentissage.

Construction Sur un ensemble d'apprentissage constitué d'images et de points de référence mis en correspondance sur chacune des images, deux statistiques sont évaluées : une statistique sur les variations possibles de forme et une statistique sur les variations possibles d'apparence.

Analyse de forme Le but de l'analyse de forme est de modéliser les variations de formes « internes » des objets de la classe étudiée. Les déformations dues aux transformations géométriques, notamment translation, rotation et changement d'échelles sont prises en compte séparément. Afin de séparer les déformations géométriques des déformations intrinsèques, il est nécessaire de normaliser géométriquement l'ensemble des N formes de la base d'apprentissage. Pour ce faire, une analyse de Procrustes est utilisée.

Chaque forme de l'ensemble d'apprentissage comprenant V points est notée :

$$\mathbf{g}_i = \begin{pmatrix} x_1^i & y_1^i \\ \vdots & \vdots \\ x_V^i & y_V^i \end{pmatrix}$$

L'analyse de Procrustes consiste à aligner une forme sur une forme de référence : il s'agit de trouver les paramètres de translation, mise à l'échelle et rotation à appliquer qui font le plus correspondre à la forme de référence.

Puis cette procédure d'alignement est appliquée à l'ensemble des formes de la base d'apprentissage par une procédure itérative (détails donnés en annexe A.2). En résultat, on obtient une forme moyenne $\bar{\mathbf{g}}_0$ et les autres formes alignées à cette moyenne.

Dans la suite, l'écriture sera simplifiée et dans les paramètres dits « de forme »seront inclus les paramètres de déformations géométriques, dont la prise en compte sera détaillée en 3.2.4 (p. 57).

Une fois les formes normalisées, notées $\bar{\mathbf{g}}_i$, il est possible d'en faire une analyse statistique. L'Analyse en Composantes Principales est utilisée dans ce but : il s'agit de connaître les axes principaux de variations de formes.

La procédure est la suivante :

1. Chaque forme $\bar{\mathbf{g}}_\mathbf{i}$ est représentée sous forme de vecteur colonne $vec(\bar{\mathbf{g}}_i)$ et stockée dans la matrice $\mathbf{G}$, qui est donc de taille $2V \times N$;
2. On construit la matrice $\mathbf{B} = \mathbf{G} - vec(\bar{\mathbf{g}}_0).\mathbf{1}_{1\times N}$; où $vec(\bar{\mathbf{g}}_0)$ est la forme moyenne et $\mathbf{1}_{1\times N}$ le vecteur ligne unitaire ;
3. La matrice de covariance est donnée par $\mathbf{C} = \frac{1}{(N-1)}\mathbf{B}\mathbf{B}^T$ de taille $2V \times 2V$;
4. Les composantes principales $\mathbf{s}_i$ de formes sont données par une décomposition en vecteurs propres. $\mathbf{C}\mathbf{s}_i = \lambda_i \mathbf{s}_i$. Les vecteurs propres principaux $\mathbf{s}_i$ correspondent aux valeurs propres λ_i les plus grandes.

Il est possible de ne retenir qu'une partie des vecteurs propres en se basant sur le critère de représentativité des vecteurs retenus. On retiendra suffisamment de vecteurs pour expliquer un certain pourcentage de la variance totale de l'ensemble d'apprentissage. La formule liant ce nombre de vecteurs n_p au pourcentage ρ de variance expliquée est :

$$\rho = \frac{\sum_{i=1}^{n_p} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

Ainsi, une forme $\mathbf{s}$ peut être définie par :

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n_p} \mathbf{p}_i^s \mathbf{s}_\mathbf{i}$$

où les $\mathbf{p}_i^s$ sont les coefficients pondérateurs des modes de variations $\mathbf{s}_i$.

ou, en notant $\mathbf{S}$ la matrice des vecteurs propres et $\mathbf{p}^s$ le vecteur des coefficients pondérateurs :

$$\mathbf{s} = \mathbf{s}_0 + \mathbf{S}\mathbf{p}^s$$

De plus, la projection d'une forme $\mathbf{s}$ dans le sous-espace propre fournit les coefficients pondérateurs :

$$\mathbf{p}^s = \mathbf{S}^T(\mathbf{s} - \mathbf{s}_0)$$

En effet, les vecteurs propres, colonnes de la matrice $\mathbf{S}$ sont orthogonaux entres eux (c'est une propriété de l'analyse en composantes principales) ; l'inverse $\mathbf{S}^{-1}$ est donc égale à la transposée $\mathbf{S}^T$

Analyse de texture L'analyse de texture consiste en une normalisation des textures de visage afin d'en faire une analyse statistique. Les textures des visages de chacune des images de la base d'apprentissage seront transformées pour qu'elles soient représentées avec des coordonnées homogènes. Pour ce faire, on choisit une forme de référence ($\mathbf{s}_0$ par exemple) et chacune des images sera déformée vers cette forme de référence (voir l'annexe A.3 pour les détails).

Une texture vectorisée peut donc être définie par :

$$\mathbf{t} = \mathbf{t}_0 + \sum_{i}^{m} \mathbf{p}_i^t \mathbf{t}_i$$

ou encore

$$\mathbf{t} = \mathbf{t}_0 + \mathbf{T}\mathbf{p}_t$$

De la même manière, les paramètres d'une texture sont donnés par :

$$\mathbf{p}_t = \mathbf{T}^T(\mathbf{t} - \mathbf{t}_0)$$

AAM combiné Il est possible de travailler avec un seul espace, combiné des deux espaces de forme et de texture. Il s'agit d'appliquer une analyse en composantes principales sur les données issues de la combinaison des données de forme et de texture. Cette construction permet d'avoir une représentation compacte de l'information et d'exprimer les corrélations qui pourraient exister entre forme et texture.

Pour chaque échantillon de la base d'apprentissage, de forme $\mathbf{s}$ et de texture $\mathbf{t}$, on construit le vecteur $\mathbf{b}$, par concaténation des paramètres de forme et de texture :

$$\mathbf{b} = \begin{bmatrix} \mathbf{W}\mathbf{p}^s \\ \mathbf{p}^t \end{bmatrix} = \begin{bmatrix} \mathbf{W}\mathbf{S}^T(\mathbf{s} - \mathbf{s}_0) \\ \mathbf{T}^T(\mathbf{t} - \mathbf{t}_0) \end{bmatrix}$$

$\mathbf{W}$ est une matrice diagonale de pondération, qui permet d'équilibrer les différences d'unités entre la forme (distance, généralement en pixels) et texture (intensité lumineuse).

Après une analyse en composantes principales, on obtient le modèle suivant :

$$\mathbf{b} = \mathbf{P}\mathbf{c}$$

Le vecteur $\mathbf{c}$, appelé aussi vecteur d'apparence, représente les coordonnées de la forme-texture $\mathbf{b}$ dans l'espace combiné $\mathbf{P}$. Il est donc possible de travailler directement dans cet espace combiné pour agir sur la forme et sur la texture.

Dans la suite, nous travaillerons sur les deux espaces de forme $\mathbf{S}$ et de texture $\mathbf{T}$, de manière indépendante.

Méthodes d'adaptation d'AAM

Nous présentons dans cette section les différentes méthodes permettant de décrire au mieux l'image d'un visage par l'instanciation d'un modèle à apparence active.

Nous présentons dans un premier temps les travaux de Cootes *et al.* qui ont été les premiers, puis d'autres approches dérivées de cette première formulation.

Dans tous les cas, le problème est vu comme une recherche des paramètres d'un modèle à apparence active effectuée itérativement en optimisant un critère de qualité d'adaptation du modèle. Les fonctions d'erreur servant à la définition de la qualité d'adaptation peuvent changer selon les auteurs, ainsi que les méthodes d'optimisation utilisées.

Méthode Cootes En supposant un modèle à apparence active avec n composantes de formes et m composantes d'apparence $\mathbf{t}_1(\mathbf{x}), \ldots, \mathbf{t}_m(\mathbf{x})$, le problème est de trouver le vecteur $\mathbf{q} \in \mathbb{R}^{m+n}$:

$$\mathbf{q} = \arg\min_{q} \sum_{\mathbf{x} \in \mathbf{s}_0} E(\mathbf{x}; \mathbf{q})^2$$

où

$$E(\mathbf{x}; \mathbf{q}) = \mathbf{t}_0(\mathbf{x}) + \sum_{i}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x}; \mathbf{p}^s))$$

et $\mathbf{q} = [\mathbf{p}^s, \mathbf{p}^t]$, concaténation des paramètres de forme et d'apparence.

La fonction $W(\mathbf{x}; q)$ représente la transformation permettant de projeter l'image $I(\mathbf{x})$ dans des coordonnées de référence (voir l'algorithme de remplissage en A.3).

Un développement de Taylor à l'ordre 1 de la fonction d'erreur donne :

$$E(\mathbf{x}; \mathbf{q} + \Delta\mathbf{q}) = E(\mathbf{x}; \mathbf{q}) + \frac{\partial E}{\partial \mathbf{q}} \Delta\mathbf{q}$$

La solution à $E(\mathbf{x}; \mathbf{q} + \Delta\mathbf{q})$ est donnée par :

$$\Delta\mathbf{q} = -\sum_{\mathbf{x}} \left[\left(\frac{\partial E}{\partial \mathbf{q}}^T \frac{\partial E}{\partial \mathbf{q}} \right)^{-1} \frac{\partial E}{\partial \mathbf{q}}^T E(\mathbf{x}; \mathbf{q}) \right] \tag{2.1}$$

notée encore

$$\Delta\mathbf{q} = -\sum_{\mathbf{x}} (\mathbf{R}(\mathbf{x}) E(\mathbf{x}; \mathbf{q}))$$

où $\mathbf{R}(\mathbf{x})$ est un vecteur de dimension $(n+m) \times 1$

Le vecteur $\frac{\partial E}{\partial \mathbf{q}}$ est supposé constant et est calculé par différenciation numérique sur un ensemble d'apprentissage.

$$\frac{\partial E}{\partial \mathbf{q}_j} = E(\mathbf{x}; \mathbf{q} + k\delta\mathbf{q}_j) - E(\mathbf{x}; \mathbf{q}), j = 1, \ldots, n+m \tag{2.2}$$

L'estimation est effectuée avec plusieurs valeurs de k, sur un ensemble d'images (faisant partie de l'ensemble d'apprentissage ou synthétisées) puis moyennée. $E(\mathbf{x}; \mathbf{q})$ représente l'image des résidus lorsque le modèle est correctement placé (elle est donc proche de l'image nulle). $E(\mathbf{x}; \mathbf{q} + \delta\mathbf{q}_j)$ représente l'image des résidus pour un modèle construit avec les paramètres optimaux $\mathbf{q}$ auxquels sont ajoutés une variation minimale sur la composante j (qui peut être une variation de forme ou d'apparence).

La procédure complète d'adaptation d'un modèle déformable à une image consiste à itérer jusqu'à convergence du modèle. Une itération comprend l'application de la formule de mise à jour 2.1 de manière pondérée par un scalaire k et à une nouvelle évaluation de l'erreur. Si la nouvelle erreur est inférieure à l'ancienne, on continue les itérations. Si la nouvelle erreur est supérieure à l'ancienne, on applique de nouveau la formule de mise à jour en diminuant le coefficient pondérateur. L'algorithme itère jusqu'à ce que l'erreur soit stabilisée.

Les limites de cette méthode sont :

1. l'approximation linéaire de l'erreur (par l'utilisation d'un développement de Taylor d'ordre 1), cette approximation est cependant acceptable par la nature itérative de l'algorithme ;
2. le fait que les gradients sont considérés constants ;
3. la méthode de calcul des gradients. En effet, bien que le gradient soit précalculé, il demande de nombreuses évaluations et la qualité de l'approximation dépend de cette étape. Ainsi, il est généralement nécessaire de prendre en considération plusieurs variations (coefficients k dans l'équation précédente) pour être capable de s'adapter sur une nouvelle image. De plus, il est aussi nécessaire de trouver des stratégies particulières pour que l'algorithme d'adaptation ne soit pas dépendant de l'arrière-plan. En effet, si l'arrière-plan lors de l'adaptation est différent de l'arrière-plan de la base d'apprentissage, il est nécessaire de générer un arrière-plan aléatoire.
4. la méthode d'optimisation, réglée par un coefficient pondérateur, qui limite l'efficacité de l'algorithme.

Méthode Ahlberg Ahlberg *et al.* [Ahlberg 01b] proposent un système utilisant un modèle de visage 3D déformable capable de s'adapter en déformations au cours d'une séquence vidéo.

Ils utilisent pour ce faire un formalisme proche des modèles à apparence active. Le modèle de forme utilisé est en trois dimensions et les modes de

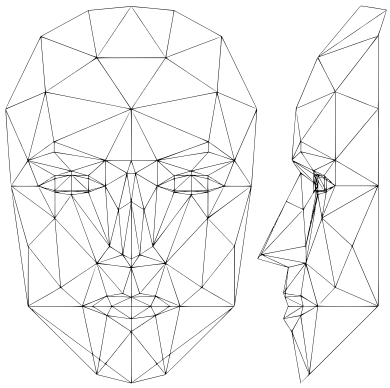


FIG. 2.13 – Modèle 3D de visage CANDIDE

déformation sont fixés par le modèle ; ils ne dérivent pas d'une analyse statistique. Il s'agit du modèle CANDIDE [Ahlberg 01a] représentant un visage générique simplifié qui peut varier selon des paramètres morphologiques ou d'expressions. Les paramètres morphologiques correspondent au FAPU de la norme MPEG-4. Les paramètres d'expression sont issus des FAP de MPEG-4 et de quelques AU de FACS.

La fonction d'erreur à minimiser n'est pas la différence entre la texture du visage observé dans l'image et une synthèse avec les paramètres actuels, mais l'erreur de reconstruction de la texture du visage. L'image d'erreur est donc, après réécriture avec nos notations :

$$E(\mathbf{x};\mathbf{p}^s) = I(W(\mathbf{x};\mathbf{p}^s)) - [\mathbf{t}_0(\mathbf{x}) + \mathbf{T}\mathbf{T}^T(I(W(\mathbf{x};\mathbf{p}^s)) - \mathbf{t}_0(\mathbf{x}))]$$

où $\mathbf{T}$ est la matrice contenant les $\mathbf{t}_i(\mathbf{x})$ en chaque colonne. Il s'agit donc de la différence entre $I(W(\mathbf{x};\mathbf{p}^s))$ et sa reconstruction sur l'espace des textures propres $\mathbf{T}$.

La solution est donnée par la même équation que dans la méthode de Cootes (équation 2.1). Ici aussi, le gradient $\frac{\partial E}{\partial \mathbf{p}^s}$ est calculé par différenciation numérique en faisant varier chacun des paramètres de forme à partir d'une position correcte du modèle.

L'utilisation d'un modèle 3D ne change pas la manière d'appliquer l'algorithme, la projection de la texture du visage est effectuée en considérant une transformation affine par morceaux du maillage 3D projeté en 2D sur un maillage 2D de référence (qui correspond ici au modèle CANDIDE neutre projeté en 2D).

La fonction d'erreur, basée sur l'erreur de reconstruction sur la base des vecteurs propres d'apparence, permet à la phase d'apprentissage d'être plus succincte, puisque les paramètres d'apparence ne font pas partie de l'optimisation : ils sont donnés par projection sur le sous-espace propre.

Cependant, la fonction d'erreur est plus coûteuse à évaluer que celle des AAM classiques, puisqu'elle contient le calcul d'une erreur de reprojection.

Pour une forme de référence échantillonnée sur N pixels et un AAM avec m vecteurs d'apparence, l'évaluation de la fonction d'erreur, en terme d'accès mémoire, est de l'ordre de mN. L'évaluation de la fonction d'erreur utilisée par Ahlberg *et al.*, si la matrice $\mathbf{T}\mathbf{T}^{\mathbf{T}}$ est précalculée, est de l'ordre de N^3.

Aucune étude comparative n'existe précisant l'impact du choix de la fonction d'erreur sur le comportement de l'AAM : vitesse de convergence, robustesse, précision, etc.

Méthode Baker & Matthews Dans [Matthews 03], les auteurs proposent de poser le problème d'adaptation d'un AAM sur une image dans le contexte de l'*alignement d'images*. Il s'agit de modifier l'algorithme de Lucas–Kanade pour le rendre efficace et applicable aux AAM. La manière de mettre à jour les paramètres (par composition et inversion) assure des conditions raisonnables permettant de considérer le gradient comme constant et rendre ainsi l'algorithme efficace. Plus de détails seront donnés dans le chapitre 3.

La fonction à minimiser est :

$$E(\mathbf{x};\mathbf{q}) = \sum_{\mathbf{x}} \left[\mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s)) \right]^2$$

Et, dans le cadre de l'algorithme à composition inverse, la fonction itérative suivante :

$$\sum_{\mathbf{x}} \left[\mathbf{t}_0(W(\mathbf{x};\Delta\mathbf{p}_s)) + \sum_{i=1}^{m} (\mathbf{p}_i^t + \Delta\mathbf{p}_i^t)\mathbf{t}_i(W(\mathbf{x};\Delta\mathbf{p}_s)) - I(W(\mathbf{x};\mathbf{p}_s)) \right]^2$$

Les paramètres de forme sont mis à jour par $W(\mathbf{x};\mathbf{q}) \leftarrow W(\mathbf{x};\mathbf{q}) \circ W(\mathbf{x};\Delta\mathbf{p})^{-1}$ et les paramètres d'apparence par $\mathbf{p}_t \leftarrow \mathbf{p}_t + \Delta\mathbf{p}_t$.

La solution est donnée à chaque itération par :

$$[\Delta\mathbf{p}_s, \Delta\mathbf{p}_t] = -H^{-1} \sum_{\mathbf{x}} G(\mathbf{x})^T E(\mathbf{x}) \tag{2.3}$$

où

$$G(\mathbf{x}) = \left[(\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i)\frac{\partial W}{\partial \mathbf{p}_1^s}, ..., (\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i)\frac{\partial W}{\partial \mathbf{p}_n^s}, \mathbf{t}_1(\mathbf{x}), ..., \mathbf{t}_m(\mathbf{x}) \right]$$

et $H = \sum_x G(\mathbf{x})^T G(\mathbf{x})$.

Comparée aux formulations précédentes, celle-ci propose un calcul analytique du gradient plutôt qu'une estimation par différenciation numérique. Ceci permet de s'affranchir des problèmes du choix des exemples à considérer

pour l'apprentissage du gradient. De plus, chaque itération correspond directement à l'évaluation de l'équation 2.3 ; il n'est plus nécessaire de faire varier un coefficient pondérateur du gradient comme précédemment.

La formulation donnée ici (nommée *simultaneous algorithm* par les auteurs) exige le calcul de $G(\mathbf{x})$ à chaque itération puisqu'il dépend des paramètres actuels d'apparence λ. C'est la formulation la plus précise et robuste. Cependant, dans certains cas, des approximations peuvent être faites et le gradient peut être précalculé. En particulier, deux variantes nommées *project-out algorithm* et *normalizing algorithm* supposent que la variation d'apparence entre deux images est faible, ce qui permet un précalcul du gradient et une grande efficacité.

La principale approximation faite concerne la transformation géométrique utilisée. En effet, la méthode est issue de l'algorithme de Lucas–Kanade qui modélise le problème en supposant une transformation géométrique W entre l'image et un modèle synthétique. Pour que la transformation géométrique W puisse être utilisée, il faut que l'inverse W^{-1} et la composition $W_1 \circ W_2$ soient définies. Or, l'inversion et la composition de transformations affines par morceaux, utilisées pour les AAM, sont mal définies ; les approximations choisies pour leur définition complète (voir 3.2.2 et 3.2.3, p. 56) sont cependant raisonnables.

3D Morphable Models Dans [Romdhani 04], les auteurs présentent un système de synthèse et d'analyse de visage, basé sur des modèles déformables 3D de visages.

L'idée est de modéliser l'ensemble des visages humains par combinaison linéaire d'un ensemble réduit de visages. L'utilisation de la 3D permet de modéliser un visage indépendamment de sa pose et de ses conditions d'éclairage. L'extraction des caractéristiques faciales se fait par comparaison de l'image étudiée avec un rendu du visage 3D avec une pose et un éclairage particuliers.

Chaque visage 3D de la base d'apprentissage a été acquis par un dispositif laser de numérisation 3D et est donc défini par une image de profondeurs dense et une image de texture.

La première étape consiste à mettre en correspondance les différentes coordonnées 3D de chacun des modèles afin de pouvoir en faire une analyse statistique correcte. Cette étape est effectuée via l'utilisation d'une technique de calcul du flux optique dense.

Une fois les données alignées, deux espaces de variations de forme et de texture sont constitués. Les formes sont, comme pour les AAM classiques, des maillages. La différence tient dans le fait que ces maillages sont en trois dimensions et qu'ils sont denses : à chaque voxel de l'image des profondeurs est associé un triangle.

Le but de l'analyse est de trouver les paramètres descripteurs de forme et de texture d'un visage observé sur une image. Il s'agit donc d'un problème de mise en correspondance 3D sur 2D.

Les auteurs ont utilisé l'algorithme *inverse compositional* appliqué à ce contexte, bien qu'il ne soit pas possible de l'étendre de manière générale à la mise en correspondance 3D sur 2D (voir [Baker 04d] pour la démonstration).

Pour pallier ce problème, les auteurs posent la fonction d'erreur à minimiser comme étant la différence pixel à pixel de la texture du modèle « déroulée » en 2D et des pixels correspondants dans l'image d'entrée, en ajoutant une transformation de l'espace 3D à l'espace 2D « déroulé ». Cependant, dans ce formalisme, la mise à jour des paramètres ne peut se faire aussi facilement que dans le cadre de l'*inverse compositional* 2D classique. En effet, il est nécessaire de pré-calculer des matrices jacobiennes pour un ensemble donné de poses et de choisir à chaque itération celle correspondant à la pose la plus proche de la pose actuelle.

L'adaptation de modèles 3D denses permet des applications intéressantes en synthèse d'image : modification de la pose 3D d'un visage à partir d'une seule photo, modification de l'éclairage, etc.

Le nombre de paramètres retenus pour permettre une adaptation à un visage quelconque étant très nombreux (entre 100 et 200 modes de variations de forme et texture sont généralement retenus par les auteurs), le modèle doit être initialisé manuellement de manière précise afin d'éviter de tomber dans des minima locaux lors de l'optimisation. De plus, le nombre important de composantes retenues et la taille du modèle de formes (comprenant plusieurs milliers de triangles) rend l'ensemble de la procédure très coûteuse en temps de calcul.

Il est cependant à noter que la base d'apprentissage utilisée par les auteurs contenant 200 visages différents ne leur a jamais posé de problèmes pour l'adaptation à des visages inconnus, et ce notamment par l'utilisation d'un modèle déformable segmenté : le visage est partitionné en quatre composantes et les statistiques de forme et de texture sont calculées indépendamment sur chacune des composantes, permettant de démultiplier le pouvoir de représentation de la base par quatre (intuitivement, il s'agit de dire qu'un visage quelconque peut être constitué des yeux de l'identité A et de la bouche de l'identité B, plutôt que du visage de A mélangé au visage de B).

Les modèles 3D déformables ont été principalement utilisés par les auteurs pour la tâche d'identification de visage. Cependant, l'analyse d'expressions a été proposée dans [Romdhani 05] en ajoutant à la base 3D des identités, une base 3D d'expressions. Les variations dues aux expressions sont supposées former un sous-espace orthogonal aux variations dues aux variations morphologiques. Cette modélisation s'avère insuffisante pour permettre un suivi précis des expressions (voir Fig. 5.9 dans [Romdhani 05] où un sourire de Duchenne est par exemple reconstruit par un sourire posé).

2.3.4 Bilan

Nous avons présenté un bref tour d'horizon des méthodes informatiques utilisées pour l'analyse du visage, et en particulier pour la description de ses expressions.

Nous avons distingué plusieurs approches : les méthodes basées sur une segmentation *a priori* du visage et les méthodes qui voient le visage dans sa globalité. La deuxième approche est généralement construite à partir d'une étape d'apprentissage, ce qui permet une certaine souplesse vis-à-vis des différents contextes possibles d'étude : nombre de visages à traiter, nombre d'expressions, type d'éclairage, etc. La construction de la base d'apprentissage est un problème à part entière puisque dans bien des cas, il est nécessaire de disposer d'une masse de données importante et d'effectuer une étape manuelle de prétraitement. Cependant, nous avons choisi de retenir les méthodes basées sur un apprentissage pour leur souplesse d'application.

Depuis les travaux de Turk *et al.* [Turk 91], une approche populaire consiste à considérer le visage comme un point dans un espace vectoriel de grande dimension, dont les axes représentent certaines déformations possibles. Avec cette représentation, il est possible d'appliquer les techniques connues d'analyse de données pour les diverses tâches d'analyse du visage : analyse en composantes principales, analyse en composantes indépendantes, analyse discriminante de Fisher, méthodes à noyaux, etc. Ceci nécessite une représentation vectorielle des visages et donc un travail de normalisation (appelé parfois *alignement*) : redimensionnement d'un ensemble d'images pour une analyse statistique de leur texture, alignement de formes, etc.

C'est pourquoi nous avons choisi d'utiliser le formalisme des modèles à apparence active (AAM). En plus des qualités mentionnées, il s'avère que les AAM sont aussi des modèles génératifs, indiquant qu'il existe une bijection entre l'espace des images de visages et l'espace des paramètres (de forme et d'apparence) de l'AAM. Ceci permet d'envisager des applications intéressantes dans le domaine de la synthèse d'images, en modifiant un visage préalablement modélisé par un AAM.

Parmi les différentes variantes existantes, nous nous sommes basés sur les travaux de Matthews & Baker qui ont formulé le problème d'adaptation d'un AAM dans un cadre mathématique rigoureux permettant d'améliorer la technique initiale et de facilement la modifier pour traiter les occultations, ajouter des *a priori* sur les paramètres de formes et/ou d'apparence, considérer un modèle 3D, considérer une caméra supplémentaire pour adapter l'AAM, etc.

Il se trouve également que ces méthodes d'adaptation d'AAM nous ont semblé, au début de cette thèse, être prometteuses et, à l'exception des travaux de leurs auteurs, peu étudiées.

La section suivante présente ainsi les algorithmes de Baker & Matthews en détail.

Deuxième partie

Suivi des déformations faciales

Chapitre 3

Algorithme à composition inverse

Dans [Matthews 03], les auteurs proposent de poser le problème d'adaptation d'un AAM sur une image dans le contexte de l'*alignement d'images*.

Dans une première série d'articles [Baker 04b, Baker 03b, Baker 03a, Baker 04a, Baker 04d], les auteurs modifient l'algorithme de Lucas - Kanade [Lucas 81], utilisé de manière classique en alignement d'images afin de le rendre efficace. Puis, cet algorithme d'alignement est appliqué aux AAM dans [Matthews 03].

Nous présentons dans un premier temps l'algorithme d'alignement d'images de Lucas–Kanade, puis comment l'appliquer au formalisme des AAM.

3.1 Algorithme de Lucas Kanade

L'algorithme de Lucas-Kanade tente de retrouver la déformation d'un modèle sur une image. On suppose ainsi qu'on observe une image d'entrée $I(\mathbf{x})$ contenant une version *déformée* d'une image modèle $M(\mathbf{x})$. Les déformations peuvent être de plusieurs types et on supposera dans un premier temps une déformation de type affine.

Les déformations possibles sont modélisées par une fonction $W(\mathbf{x}; \mathbf{p}^s)$ où $\mathbf{p}^s$ est un vecteur représentant les paramètres de la déformation appliquée en chaque pixel $\mathbf{x}$. La fonction W transforme chaque coordonnée de l'image modèle $M(\mathbf{x})$ en une coordonnée de l'image $I(\mathbf{x})$.

Le problème consiste à minimiser la fonction d'erreur :

$$\sum_{\mathbf{x} \in M} [I(W(\mathbf{x}; \mathbf{p}^s)) - M(\mathbf{x})]^2$$

Il s'agit d'une optimisation non-linéaire. En effet les valeurs des pixels de $I(\mathbf{x})$ ne sont pas fonction de $\mathbf{x}$ dans le cas général. Ainsi, on considère une

formulation où les paramètres seront optimisés itérativement avec :

$$\sum_{\mathbf{x}} [I(W(\mathbf{x}; \mathbf{p}^s + \Delta\mathbf{p}^s)) - M(\mathbf{x})]^2 \tag{3.1}$$

A chaque itération, les paramètres seront mis à jour par :

$$\mathbf{p}^s \leftarrow \mathbf{p}^s + \Delta\mathbf{p}^s$$

L'équation 3.1 peut être approximée par un développement de Taylor du premier ordre en $W(\mathbf{x}; \mathbf{p}^s)$ par :

$$\sum_{\mathbf{x}} \left[I(W(\mathbf{x}; \mathbf{p}^s)) + \nabla I \frac{\partial W}{\partial \mathbf{p}^s} \Delta\mathbf{p}^s - M(\mathbf{x}) \right]^2 \tag{3.2}$$

où ∇I représente le gradient de l'image I évalué en $W(\mathbf{x}; \mathbf{p}^s)$ et $\frac{\partial W}{\partial \mathbf{p}^s}$ représente la matrice des dérivées partielles (la jacobienne) de la transformation W (évaluée aussi en $W(\mathbf{x}; \mathbf{p}^s)$).

En dérivant par rapport à $\Delta\mathbf{p}^s$, suivant les règles de dérivation matricielles classiques [Fang 90], on obtient :

$$\sum_{\mathbf{x}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}^s} \right]^T [I(W(\mathbf{x}; \mathbf{p}^s) + \nabla I \frac{\partial W}{\partial \mathbf{p}^s} \Delta\mathbf{p}^s - M(\mathbf{x})]$$

En posant cette dernière équation égale à 0 (condition nécessaire d'un minimum), on obtient la solution.

$$\Delta\mathbf{p}^s = -H^{-1} \sum_{\mathbf{x}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}^s} \right]^T [M(\mathbf{x}) - I(W(\mathbf{x}; \mathbf{p}^s))]$$

où

$$H = \sum_{\mathbf{x}} \left[\nabla I \frac{\partial W}{\partial \mathbf{p}^s} \right]^T \left[\nabla I \frac{\partial W}{\partial \mathbf{p}^s} \right]$$

A chaque itération, l'image gradient ∇I, la jacobienne $\frac{\partial W}{\partial \mathbf{p}^s}$ et donc la matrice H doivent être recalculées. Ainsi, cette première version manque d'efficacité.

3.1.1 Composition inverse

Afin de pallier à ce problème, les auteurs proposent dans [Baker 04b] une variante de l'algorithme de Lucas–Kanade, appelée *inverse compositional*.

Considérons dans un premier temps la variante « à composition », où la fonction d'erreur est :

$$\sum_{\mathbf{x}} [M(\mathbf{x}) - I(W(W(\mathbf{x}; \Delta\mathbf{p}^s); \mathbf{p}^s))]^2$$

Le principe est ici de mettre à jour les paramètres par une composition :

$$W(\mathbf{x};\mathbf{p}^s) \leftarrow W(\mathbf{x};\mathbf{p}^s) \circ W(\mathbf{x};\Delta\mathbf{p}^s)$$

L'idée est donc de trouver à chaque itération un $\Delta\mathbf{p}^s$ qui, considéré comme paramètre d'une transformation W appliquée à I, permet lorsque celle-ci est composée avec la transformation actuelle, de minimiser l'erreur.

Le problème peut être posé en cherchant un $\Delta\mathbf{p}^s$ qui, considéré comme paramètre d'une transformation W appliquée à l'image modèle M (et non plus à I), permet lorsque celle-ci est composée avec la transformation actuelle, de minimiser l'erreur.

Il s'agit dans ce cas, de minimiser l'erreur suivante :

$$\sum_{\mathbf{x}} [M(W(\mathbf{x};\Delta\mathbf{p}^s)) - I(W(\mathbf{x};\mathbf{p}^s))]^2 \tag{3.3}$$

Et les paramètres sont mis à jour par :

$$W(\mathbf{x};\mathbf{p}^s) \leftarrow W(\mathbf{x};\mathbf{p}^s) \circ W(\mathbf{x};\Delta\mathbf{p}^s)^{-1}$$

Cette nouvelle formulation du problème permet alors d'avoir une résolution bien plus efficace que la formulation initiale, comme le détaille la démonstration suivante.

Soit $F(\mathbf{x};\Delta\mathbf{p}^s) = M(W(\mathbf{x};\Delta\mathbf{p}^s)) - I(W(\mathbf{x};\mathbf{p}^s))$. En utilisant un développement de Taylor du premier ordre en $W(\mathbf{x};0)$ (et non plus en $W(\mathbf{x};\mathbf{p}^s)$ comme précédemment), l'expression 3.3 devient :

$$\sum_{\mathbf{x}} \left[F(\mathbf{x};0) + \frac{\partial F}{\partial \mathbf{p}^s}\Delta\mathbf{p}^s\right]^2 = \sum_{\mathbf{x}} \left[M(W(\mathbf{x};0)) + \nabla M \frac{\partial W}{\partial \mathbf{p}^s}\Delta\mathbf{p}^s - I(W(\mathbf{x};\mathbf{p}^s))\right]^2$$

En posant la transformation $W(\mathbf{x};0)$ comme étant équivalente à la fonction identité, on obtient :

$$\sum_{\mathbf{x}} \left[M(\mathbf{x}) + \nabla M \frac{\partial W}{\partial \mathbf{p}^s}\Delta\mathbf{p}^s - I(W(\mathbf{x};\mathbf{p}^s))\right]^2$$

En suivant la même procédure de dérivation que précédemment, la solution est donnée par (les détails des calculs peuvent aussi être trouvés dans [Baker 04b]) :

$$\Delta\mathbf{p}^s = -H^{-1} \sum_{\mathbf{x}} \left[\nabla M \frac{\partial W}{\partial \mathbf{p}^s}\right]^T [M(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s))]$$

où

$$H = \sum_{\mathbf{x}} \left[\nabla M \frac{\partial W}{\partial \mathbf{p}^s}\right]^T \left[\nabla M \frac{\partial W}{\partial \mathbf{p}^s}\right]$$

∇M ne dépend pas de $\mathbf{p}^s$ et peut donc être précalculée. De même, la jacobienne $\frac{\partial W}{\partial \mathbf{p}^s}$ est maintenant calculée en $W(\mathbf{x};0)$ et H peut donc être précalculée.

La solution $\Delta\mathbf{p}^s$ ne peut être cependant appliquée directement. La nouvelle transformation est donnée par $W(\mathbf{x};\mathbf{p}^s) \leftarrow W(\mathbf{x};\mathbf{p}^s) \circ W(\mathbf{x};\Delta\mathbf{p}^s)^{-1}$.

3.1.2 Variations de texture

L'algorithme de Lucas-Kanade et sa variante *inverse compositional* considèrent uniquement une transformation géométrique entre l'image d'entrée $I(\mathbf{x})$ et l'image modèle $M(\mathbf{x})$. Il est possible d'étendre ces algorithmes pour qu'ils prennent en compte à la fois une déformation géométrique et des variations photométriques (ou variations de texture).

En particulier, l'image modèle peut être représentée sous la forme :

$$M(\mathbf{x}) = \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x})$$

où les $\mathbf{t}_i(\mathbf{x})$ sont des modes de variation d'apparence et les $\mathbf{p}_i^t$ des coefficients pondérateurs de ces variations.

Ainsi, l'image des résidus devient :

$$E(\mathbf{x}) = \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x}; \mathbf{p}^s)) \tag{3.4}$$

Et, dans le cadre de l'algorithme à composition inverse, il s'agit d'optimiser la fonction itérative suivante :

$$\sum_{\mathbf{x}} \left[\mathbf{t}_0(W(\mathbf{x}; \Delta\mathbf{p}^s)) + \sum_{i=1}^{m} (\mathbf{p}_i^t + \Delta\mathbf{p}_i^t) \mathbf{t}_i(W(\mathbf{x}; \Delta\mathbf{p}^s)) - I(W(\mathbf{x}; \mathbf{p}^s)) \right]^2$$

L'optimisation est menée à la fois sur les paramètres de forme et sur les paramètres de texture, c'est pourquoi cet algorithme est nommé par les auteurs *simultaneous algorithm* (voir [Baker 03a], section 3.1). Les paramètres de forme sont mis à jour par $W(\mathbf{x}; \mathbf{p}^s) \leftarrow W(\mathbf{x}; \mathbf{p}^s) \circ W(\mathbf{x}; \Delta\mathbf{p}^s)^{-1}$ et les paramètres d'apparence par $\mathbf{p}^t \leftarrow \mathbf{p}^t + \Delta\mathbf{p}^t$.

La solution est donnée à chaque itération par :

$$[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t] = -H^{-1} \sum_{\mathbf{x}} G(\mathbf{x})^T E(\mathbf{x}) \tag{3.5}$$

où

$$G(\mathbf{x}) = \left[(\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i) \frac{\partial W}{\partial \mathbf{p}_1^s}, ..., (\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i) \frac{\partial W}{\partial \mathbf{p}_n^s}, \mathbf{t}_1(\mathbf{x}), ..., \mathbf{t}_m(\mathbf{x}) \right] \tag{3.6}$$

et $H = \sum_{\mathbf{x}} G(\mathbf{x})^T G(\mathbf{x})$.

Les différentes variantes de ces algorithmes ne sont pas détaillées ici. On pourra se référer à [Baker 04b, Baker 03a] pour une description détaillée. Baker & Matthews utilisent principalement une variante nommée *project out* dans leurs travaux sur les AAM mettant en relief son efficacité (en annonçant un algorithme de suivi en temps réel). Nous avons préféré choisir le *simultaneous* pour sa précision, au détriment d'une perte d'efficacité.

3.2 Application aux AAM

Dans [Matthews 03], les auteurs appliquent leur méthode générique d'alignement d'images *inverse compositionnal* au formalisme des AAM. Il s'agit en particulier de choisir la fonction W comme étant une transformée affine par morceaux.

3.2.1 Transformée affine par morceaux

Un modèle déformable de visage est construit de la manière suivante :

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} \mathbf{p}_i^s \mathbf{s}_i$$

où $\mathbf{s}_0$ est la forme moyenne, les $\mathbf{s}_i$ sont des vecteurs de déformations et $\mathbf{p}_i^s$ des coefficients pondérateurs.

Ainsi, une instance de la forme d'un visage peut être représentée par la fonction $\mathbf{s} = AAM(\mathbf{p}^s; \mathbf{s}_0; \mathbf{s}_i)$.

Afin de construire une fonction d'erreur, il est nécessaire de comparer l'apparence de l'image d'entrée à l'image modèle. Dans le cas des modèles déformables, il s'agit de comparer les pixels dans des coordonnées communes : ce sont par exemple l'ensemble des pixels présents à l'intérieur (dans l'enveloppe convexe) de la forme moyenne. Ainsi, la fonction d'erreur doit transformer les coordonnées des pixels de l'image d'entrée qui font partie de l'instance actuelle du modèle déformable en des coordonnées de la forme moyenne de manière à les comparer au visage modèle.

Pour ce faire, on considère qu'entre deux instances d'un modèle de forme, il existe une transformée affine de chacun de leurs triangles. L'inverse est cependant faux : une transformée affine appliquée à chaque triangle d'un modèle de forme ne résulte pas en un modèle de forme valide, puisqu'il est possible que la connexité entre les triangles soit perdue.

Ainsi, la fonction géométrique W est une transformation affine par morceaux qui transforme chaque triangle d'un modèle de forme s en un triangle correspondant de $\mathbf{s}_0$. Cette transformation $W(\mathbf{x}; \mathbf{p})$ est paramétrée par le vecteur $\mathbf{p}$ qui correspond aux coefficients pondérateurs du modèle de forme.

Avec une écriture homogène, la fonction d'erreur pour un AAM sans variation d'apparence, dans le cas de l'algorithme *inverse compositional* est :

$$\sum_{\mathbf{x} \in \mathbf{s}_0} \left[\mathbf{t}_0(W(\mathbf{x}; \Delta\mathbf{p}^s)) + I(W(\mathbf{x}; \mathbf{p}^s)) \right]^2$$

avec l'abus de notation $\mathbf{x} \in \mathbf{s}_0$ indiquant l'ensemble des coordonnées de l'enveloppe convexe de $\mathbf{s}_0$.

3.2.2 Inverse

Les bases de l'algorithme *inverse compositional* sont l'inversion et la composition présentes dans la formule de mise à jour des paramètres :

$$W(\mathbf{x};\mathbf{p}^s) \leftarrow W(\mathbf{x};\mathbf{p}^s) \circ W(\mathbf{x};\Delta\mathbf{p}^s)^{-1}$$

Il est donc nécessaire de pouvoir inverser une transformée affine par morceaux et d'en composer deux.

Une transformée affine transforme (x, y) en (x', y') par :

$$\begin{aligned} x' &= a_1 x + a_2 y + b_1 \\ y' &= a_3 x + a_4 y + b_2 \end{aligned}$$

ou encore sous forme matricielle par $\boldsymbol{x'} = A\boldsymbol{x}$ avec :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

L'inverse d'une composition affine est défini par la matrice A^{-1}. Il est possible d'utiliser une approximation de cette transformée inverse. En effet, la transformée affine d'un triangle peut être vue comme trois translations appliquées en chacun des sommets du triangle. Par exemple, si on applique la transformée affine W sur un triangle $T1$, on obtient le triangle $T2$. La transformation peut aussi être exprimée comme étant trois vecteurs de translation dT appliqués en chacun des sommets de $T1$. L'antécédent de $T1$, obtenu en y appliquant la transformée inverse W^{-1} peut être approximé par le triangle obtenu par translation inverse $-dT$ de chacun de ses sommets (voir Fig. 3.1).

L'inverse d'une transformation affine par morceaux appliquée à un maillage doit retourner un maillage. Or, appliquer la transformée inverse en chacun des triangles d'un maillage ne retourne pas nécessairement un maillage, puisque la connexité peut être perdue.

Ainsi, l'approximation qui consiste à estimer l'inverse $\mathbf{W}^{-1}$ par translation opposée, est utilisée pour l'algorithme *inverse compositional*. En effet, le fait de traiter une transformation affine comme le déplacement des sommets d'un triangle, plutôt que comme une transformation globale de triangle, permet d'assurer le maintien de la connexité d'un maillage.

3.2.3 Composition

La composition de deux fonctions affines définies par les matrice A et B revient à appliquer la fonction affine définie par $C = AB$.

Or, dans le cas de maillages, composer deux transformations affines en chaque triangle fait perdre la connexité du maillage. Pour remédier à ce problème, il est possible là encore d'envisager une approximation qui conserve la connexité.

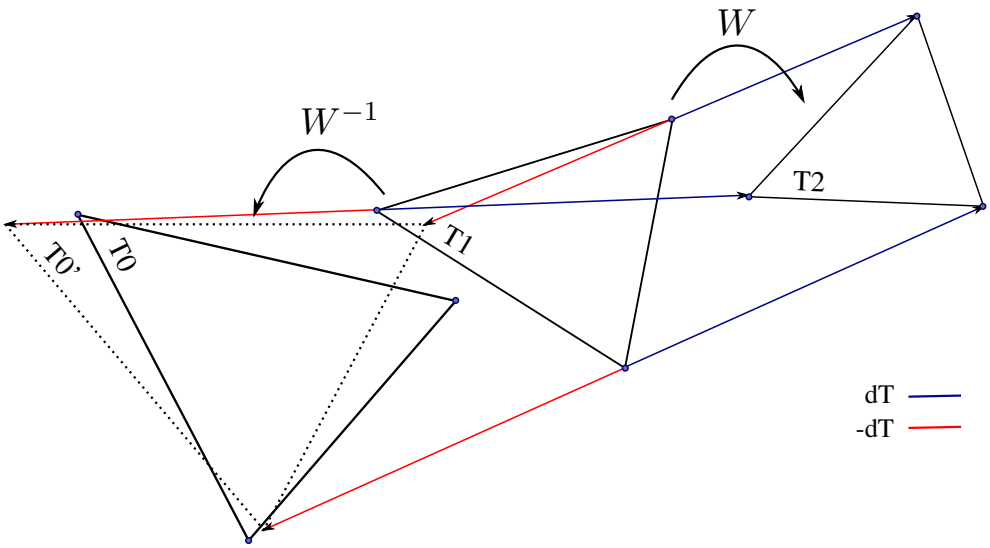


FIG. 3.1 – Illustration de l'approximation de la transformation affine inverse. L'application W transforme le triangle $T1$ en $T2$; les vecteurs de déformation équivalents dT appliqués en sens opposé sur $T1$ donnent le triangle $T0'$ qui n'est pas équivalent à l'antécédent $T0$ de $T1$ par W.

Il s'agit d'une approximation générique qui permet de maintenir la connexité d'un maillage après une transformation qui pourrait la faire perdre. La transformation est appliquée en chaque triangle du maillage d'origine ; l'ensemble des triangles après transformation ne forme plus un maillage. Les sommets du nouveau maillage sont déterminés en faisant la moyenne des coordonnées de chacun des sommets transformés (voir Fig. 3.2).

Dans l'algorithme *inverse compositional*, il est nécessaire de composer la transformation de l'estimation actuelle de la forme $W(\mathbf{x};\mathbf{p}^s)$ avec l'application transformant $\mathbf{s}_0$ en $\mathbf{s}_0 - \mathbf{S}\Delta\mathbf{p}^s$. Cette dernière est donc appliquée à la forme $\mathbf{s}$ en procédant par moyennage des sommets obtenus pour contraindre la connexité.

Les approximations faites sur l'opération d'inversion et de composition font qu'une forme obtenue après ces transformations peut ne plus faire partie de la statistique de formes et amener dans certains cas à la divergence de l'algorithme. Ainsi, il est préférable de contraindre la nouvelle forme en la reprojetant sur le sous-espace des formes apprises. La forme $\mathbf{s}$ reprojetée est alors égale à : $\mathbf{S}(\mathbf{S}^T(\mathbf{s}-\mathbf{s}_0)) + \mathbf{s}_0$.

3.2.4 Similarités euclidiennes

Les sections précédentes considéraient que le modèle n'était déformable que selon des déformations intrinsèques du modèle, dues aux différentes expressions ou à la morphologie associée à chaque identité. En réalité, le visage observé dans l'image a une position, une rotation et une échelle particulières alors que la procédure d'apprentissage a éliminé toutes ces variations géométriques

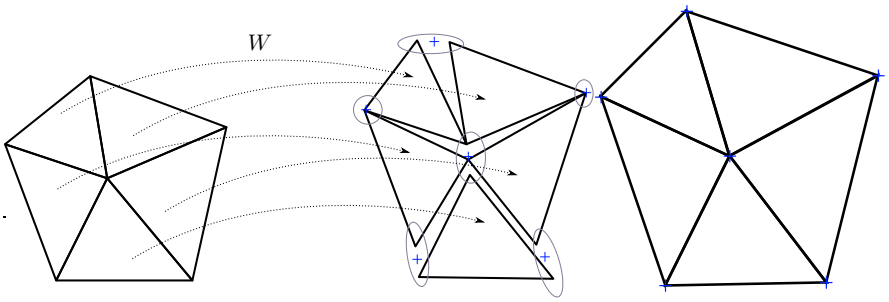


FIG. 3.2 – Illustration de la procédure de maintien de la connexité d'un maillage après transformation. La transformation W appliquée en chacun des triangles du modèle donne un ensemble de triangles non connexes. Le maillage résultat (b) est déterminé en prenant le barycentre (croix bleues) des sommets de chacun des triangles transformés.

possibles.

Pour prendre en compte ces déformations géométriques, faisant partie des similarités euclidiennes, on les considère de la même façon que les autres déformations locales. Ainsi, les similarités euclidiennes seront codées par des vecteurs de variation de forme de la même nature que les vecteurs de déformations issus de la statistique.

La translation en x, notée $\mathbf{s}_1^\star$ est un vecteur comprenant des 1 sur sa composante x et des 0 sinon.

La translation en y, notée $\mathbf{s}_2^\star$ est un vecteur comprenant des 1 sur sa composante y et des 0 sinon.

La mise à l'échelle, notée $\mathbf{s}_3^\star$ est représentée par la forme moyenne $\mathbf{s}_0$

La rotation, notée $\mathbf{s}_4^\star$ est représentée par la forme moyenne ayant subi une rotation à 90° : $(-y_1, x_1, \ldots, -y_V, x_V)$

Une forme est donc obtenue par :

$$\mathbf{s} = \mathbf{s}_0 + \sum_{i=1}^{n} \mathbf{p}^s \mathbf{s}_i + \sum_{i=1}^{4} \mathbf{p}^g \mathbf{s}_i^\star$$

ou bien encore :

$$\mathbf{s} = \mathbf{s}_0 + \mathbf{S}\mathbf{p}^s + \mathbf{S}^\star \mathbf{p}^g$$

Les paramètres de formes et les paramètres géométriques peuvent être obtenus par :

$$(\mathbf{p}^s, \mathbf{p}^g) = [\mathbf{S}|\mathbf{S}^\star]^T (\mathbf{s} - \mathbf{s}_0)$$

Cette formule est valable quand les vecteurs $\mathbf{s}_i$ et $\mathbf{s}_i^\star$ sont orthogonaux. Si ce n'est pas le cas, il est possible de lancer une procédure d'orthogonalisation de Gram–Schmidt. Une autre solution consiste à utiliser la pseudo-inverse

de la matrice $[\mathbf{S}|\mathbf{S}^\star]$ plutôt que sa transposée (cette dernière, bien que non orthogonale, a généralement un rang égal à $(n+4)$, n étant le nombre de vecteurs de variations de forme et 4 le nombre de déformations géométriques, pour rappel).

Dans la suite, les paramètres de forme sont considérés comme incluant les paramètres géométriques, sauf mention contraire.

3.2.5 Détails de calculs

Dans cette section, nous détaillons le calcul de la jacobienne $\frac{\partial W}{\partial \mathbf{p}^s}$, qui correspond à la matrice des dérivées partielles des transformations affines W par rapport aux paramètres p, ainsi que le calcul du gradient image $\nabla I(\mathbf{x})$.

Si une forme $\mathbf{s}$ est représentée par le vecteur $\mathbf{s} = (x_1, y_1, \ldots, x_V, y_V)$, il est possible d'appliquer la règle de dérivation croisée suivante :

$$\frac{\partial W(\mathbf{x};\mathbf{p}^s)}{\partial \mathbf{p}^s} = \sum_{j=1}^{V} \left[\frac{\partial W(\mathbf{x};\mathbf{p}^s)}{\partial x_j} \frac{\partial x_j}{\partial \mathbf{p}^s} + \frac{\partial W(\mathbf{x};\mathbf{p}^s)}{\partial y_j} \frac{\partial y_j}{\partial \mathbf{p}^s} \right]$$

Le premier terme $\frac{\partial W(\mathbf{x};p)}{\partial x_j}$ correspond à la variation de la destination de la transformée W lorsque la coordonnée x du sommet j varie. En reprenant les équations A.2 et A.3, et en les dérivant par rapport à x_j et y_j, on obtient, en chacune des coordonnées $\mathbf{x}$ de la forme moyenne :

$$\frac{\partial W(\mathbf{x};\mathbf{p}^s)}{\partial x_j} = (1 - \alpha(\mathbf{x}) - \beta(\mathbf{x}), 0)$$

et

$$\frac{\partial W(\mathbf{x};\mathbf{p}^s)}{\partial y_j} = (0, 1 - \alpha(\mathbf{x}) - \beta(\mathbf{x}))$$

Le deuxième terme $\frac{\partial x_j}{\partial \mathbf{p}^s}$ correspond à la variation de la coordonnée x du sommet j lorsque chacun des paramètres de forme (dans p) varie. Il s'agit donc des éléments correspondant au sommet j des vecteurs de déformations, soit $(\mathbf{S}_{j,1}, \ldots, \mathbf{S}_{j,n})$.

La jacobienne $\frac{\partial W}{\partial \mathbf{p}^s}$ est définie en chaque pixel de la forme de référence $\mathbf{s}_0$ par une matrice de taille $2 \times n$.

Le terme ∇I correspond au gradient de l'image I. Il existe plusieurs façons de le calculer. Dans notre cas, il est calculé par :

$$\frac{\partial I}{\partial x}(x,y) = \frac{1}{2}(I(x+1,y) - I(x-1,y))$$

$$\frac{\partial I}{\partial y}(x,y) = \frac{1}{2}(I(x,y+1) - I(x,y-1))$$

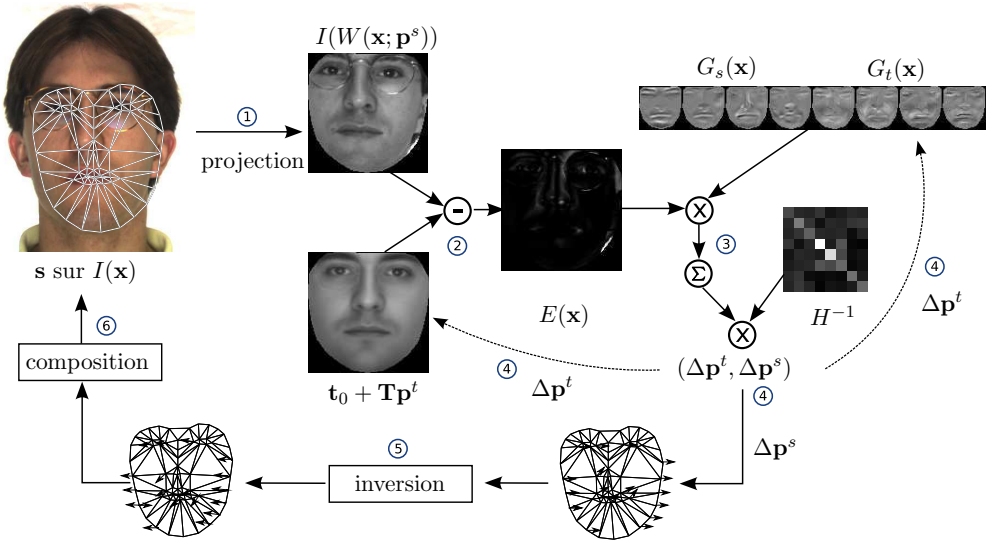


FIG. 3.3 – Principe de fonctionnement de l'algorithme *simultaneous*.

3.2.6 Algorithme

La figure 3.3 donne une vue schématique des différentes étapes de calcul pour chaque itération de l'algorithme.

L'algorithme est initialisé avec une première estimation du modèle de forme $\mathbf{s}$ sur l'image d'entrée $I(\mathbf{x})$ (en haut à gauche de la figure) et une première estimation de la texture (en haut à droite de la figure). Les paramètres de texture $\mathbf{p}^t$ (qui sont nuls généralement à la première itération) servent à la construction de l'estimation actuelle de la texture et rentrent aussi en jeu dans le calcul du gradient $G(\mathbf{x})$ et de la matrice H.

En ①, la texture à l'intérieur de chaque triangle du modèle de forme est extraite de l'image d'entrée I et transformée vers la forme moyenne $\mathbf{s}_0$ de manière à former l'image $I(W(\mathbf{x};\mathbf{p}^s))$.

En ②, la texture extraite de I, calculée à l'étape précédente et l'estimation actuelle de la texture $\mathbf{t}_0 + \mathbf{T}\mathbf{p}^t$ sont soustraites de manière à former l'image des résidus $E(\mathbf{x})$.

En ③, l'image des résidus est multipliée pixel à pixel avec $G(\mathbf{x})$. Le tout est sommé et il en résulte un vecteur de $(n+m)$ éléments qui, multiplié par l'inverse de H donnera en ④ le vecteur de mise à jour des paramètres $[\Delta\mathbf{p}^t, \Delta\mathbf{p}^t]$ (voir l'équation 3.5).

La mise à jour des paramètres de texture $\Delta\mathbf{p}^t$ servira à faire évoluer l'estimation actuelle de la texture et le gradient $G(\mathbf{x})$ à la prochaine itération.

Le vecteur $\Delta\mathbf{p}^t$ donne les modifications à apporter à $\mathbf{s}_0$ pour minimiser l'erreur.

On procède alors à l'opération d'inversion en ⑤ et à la composition en ⑥ de manière à obtenir la nouvelle forme.

Soit $\mathbf{s}$ l'estimation actuelle de la forme
Soit $\mathbf{p}^t$ l'estimation actuelle des paramètres de texture
Soit $I(\mathbf{x})$ l'image d'entrée
Itérer
    Soit $I(W(\mathbf{x};\mathbf{p}^s))$ la projection de $I(\mathbf{x})$ sur $\mathbf{s}_0$ par rapport à $\mathbf{s}$ (algorithme A.3)
    Calculer l'image des résidus $E(\mathbf{x};\mathbf{q})$ (équation 3.4)
    Calculer $G(\mathbf{x})$ et $H(\mathbf{x})$ (équation 3.6)
    Calculer $[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t]$ (équation 3.5)
    Calculer $\mathbf{s}_0 - \mathbf{S}\Delta\mathbf{p}^s$
    Calculer la nouvelle forme $\mathbf{s}$ par composition (voir 3.2.3)
    Mettre à jour $\mathbf{p}^t \leftarrow \mathbf{p}^t + \Delta\mathbf{p}^t$
    (Reprojeter la nouvelle forme sur l'espace des formes)
Fin

FIG. 3.4 – L'algorithme *simultaneous inverse compositional* appliqué aux AAM en considérant les variations de texture

Chapitre 4

Composition inverse : évaluation

Dans ce chapitre, l'algorithme à composition inverse, présenté précédemment est évalué. Il s'agit de déterminer quand une convergence acceptable de l'algorithme est atteinte, la précision atteignable par un tel algorithme, ses performances à l'égard de visages qui n'appartiennent pas à la base d'apprentissage et sa capacité à traiter le cas des rotations hors-plan et les différents types d'éclairage de la scène.

4.1 Convergence

L'algorithme à composition inverse utilise une méthode d'optimisation locale. Et de ce fait, l'algorithme ne peut converger que vers un minimum local de la fonction objectif $\sum_{\mathbf{x}} E(\mathbf{x})^2$.

Rien n'indique alors que le minimum obtenu après convergence soit le minimum global. Dans le cas où l'algorithme est lancé sur une image qui peut être entièrement expliquée par les vecteurs de déformations et de variations de texture (c'est le cas d'une image de la base d'apprentissage si le modèle a été construit avec 100% de variance de forme et de texture ou encore d'une image artificielle générée à partir de la statistique de forme et de texture), le minimum global est atteint quand l'image des résidus est nulle. Dans le cas où l'AAM ne peut pas expliquer entièrement l'observation, le minimum global de la fonction objectif ne correspond pas forcément à la configuration du modèle de forme qui aurait été obtenue par une annotation manuelle.

Ainsi, on peut distinguer plusieurs comportements : la convergence amenant à une configuration du modèle de forme « satisfaisante », la convergence amenant à un configuration non satisfaisante, et la divergence.

Une convergence est satisfaisante quand le modèle se stabilise sur une configuration qui aurait été donnée par une annotation manuelle.

Automatiser la détection d'une convergence satisfaisante n'est donc pas *a*

priori possible. Il est cependant possible de détecter les très mauvaises convergences en se basant sur une analyse statistique. L'idée est alors de supposer que la base d'apprentissage servant à la construction de l'AAM présente une vue d'ensemble des déformations réalistes et de tester une configuration par rapport à sa possibilité de réalisation.

Après avoir calculé l'écart-type de chacun des paramètres de forme σ_i, une divergence peut être détectée si :

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{p}_i^s|}{\sigma_i} > \rho_1 \text{ ou } \max_{i=1,\ldots,n}\left\{\frac{|\mathbf{p}_i^s|}{\sigma_i}\right\} > \rho_2$$

Les seuils ρ_1 et ρ_2 sont déterminés empiriquement (nous utilisons $\rho_1 = 2.5$ et $\rho_2 = 7.0$). Le membre gauche de l'équation précédente permet de détecter une divergence quand l'ensemble des paramètres est peu réaliste et le membre droit permet de détecter quand un seul des paramètres est très peu réaliste.

4.2 Précision

Dans cette section, nous évaluons la précision atteignable par l'algorithme à composition inverse. Pour ce faire, il est nécessaire de définir au préalable une mesure de précision.

De manière classique, la précision atteinte se calcule comme étant la distance à une vérité terrain. Le problème revient donc à définir la vérité terrain la plus correcte possible. Dans les cas des AAM, la vérité terrain est définie par une forme *i.e.*, un ensemble de coordonnées 2D qui correspondent aux points d'intérêts du modèle.

Cependant, les points d'intérêt ne sont pas toujours possibles à localiser précisément (au pixel près) sur une image de visage, en particulier si les points d'intérêt ne sont pas définis par de forts contrastes et ce, même pour un opérateur humain. Le problème est alors de décider parmi plusieurs annotations laquelle est la meilleure. Il n'est évidemment pas possible de décider objectivement entre plusieurs annotations. En revanche, si plusieurs annotations d'un même visage existent, il est possible d'en tirer avantage par une analyse statistique.

Chacune des différentes annotations manuelles d'un même visage comprend un bruit dans la localisation de chacun des points du modèle de forme. Afin de limiter le bruit introduit par l'annotation manuelle, la vérité terrain peut être définie comme étant les coordonnées moyennes parmi toutes les annotations de chaque point d'intérêt.

Si n_L annotations manuelles, définies pour un modèle de forme à n_V points, sont disponibles pour chacune des n_I images de visages, alors le point v de la

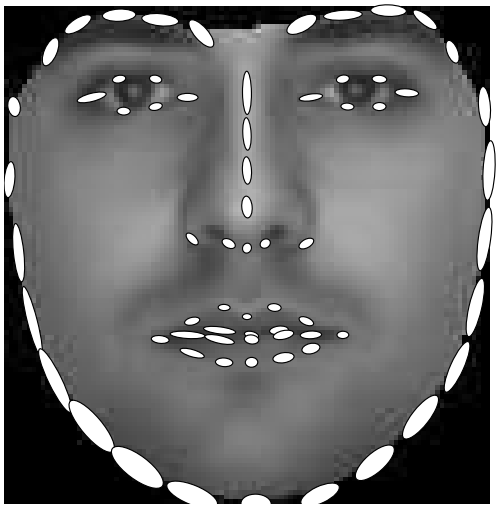


FIG. 4.1 – Représentation de la covariance de chaque point du modèle de forme (ici la forme moyenne) par des ellipses de dispersion.

forme de référence du visage i est définie par :

$$\boldsymbol{\mu}_{i,v} = \frac{1}{n_L} \sum_{l=1}^{n_L} \boldsymbol{x}_{i,v,l}$$

De plus, cette statistique permet de définir une covariance pour chacun des points du modèle :

$$\boldsymbol{\Sigma}_v = \frac{1}{n_I n_L - 1} \sum_{i=1}^{n_I} \sum_{l=1}^{n_L} (\boldsymbol{x}_{i,v,l} - \boldsymbol{\mu}_{i,v})^T (\boldsymbol{x}_{i,v,l} - \boldsymbol{\mu}_{i,v})$$

Une telle statistique a été évaluée dans [Mercier 06] sur un sous ensemble de la base de visages AR [Martinez 98], contenant 40 images (réduites en niveaux de gris) de visages d'identités différentes, vus de face, affichant l'expression neutre, et photographiés dans des conditions d'éclairage fixes. Pour chaque visage, $n_L = 10$ annotations manuelles ont été effectuées, permettant de définir une forme moyenne et une covariance pour chaque point du modèle.

Ainsi il apparaît clairement que certains points du modèle sont mieux définis que d'autres. En particulier, les points du contour du visage, puisque difficiles à localiser par des contrastes forts, sont localisés de manière imprécise, alors que les points du contour des lèvres ou des yeux sont localisés bien plus précisément.

Il est alors possible de définir une mesure de précision d'une annotation en utilisant cette information : l'objectif est de juger une annotation, obtenue manuellement ou automatiquement, par rapport au bruit introduit par les annotations manuelles de référence. Ainsi, si $\boldsymbol{s}$ est une annotation du visage i,

sa distance à la vérité terrain peut être définie par une distance point à point pondérée :

$$e(\boldsymbol{s}) = \frac{1}{n_V} \sum_{v=1}^{n_V} \sqrt{(\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})^T \boldsymbol{\Sigma}_v^{-1} (\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})}$$

Ce qui correspond à la moyenne des distances de Mahalanobis sur chaque point du modèle de forme.

4.2.1 Précision sur cas connu et inconnu

Afin de déterminer la précision atteignable par l'algorithme d'adaptation d'AAM, il est nécessaire de distinguer deux tests : un test effectué sur l'image d'un visage qui fait partie de la base d'apprentissage du modèle (cas connu) et un test effectué sur l'image d'un visage qui ne fait pas partie de la base d'apprentissage du modèle (cas inconnu).

Nous avons évalué l'algorithme simultané dans ces deux cas. Dans le cas connu, l'AAM est construit à partir du sous-ensemble de 40 images de la base AR utilisé précédemment. Pour le cas inconnu, pour chaque image de test, l'AAM a été construit sur les 39 autres images (test dit du *leave-one-out*). Nous avons retenu $n = 24$ vecteurs de variation de forme et $m = 30$ vecteurs de variation de texture, expliquant 95% de la variance totale dans chacun des cas.

L'algorithme simultané a été lancé pendant 50 itérations sur chacune des 40 images. L'initialisation a été donnée de la manière suivante : la forme vérité a été projetée sur l'espace des déformations géométriques et de formes ($[\mathbf{S}, \mathbf{S}^\star]$) et les paramètres de formes $\mathbf{p}^s$ ont été annulés. L'AAM est donc initialisé par la forme moyenne qui a subi les déformations géométriques qui l'amènent la plus proche possible de la forme vérité. Ceci permet de simuler le résultat qui serait obtenu par un détecteur de visages. Les paramètres de texture ont été initialisés à zéro.

A chaque itération la distance à la solution $e(\boldsymbol{s})$ a été enregistrée et la figure 4.2 présente la distance pour chaque itération en moyenne sur les 40 images. Sont de plus représentées en pointillés la pire, la meilleure et la précision moyenne obtenues manuellement.

Ainsi, il apparaît que l'algorithme simultané permet d'atteindre une très bonne précision dans le cas connu.

Concernant le cas inconnu les résultats sont moins bons que dans le cas connu, mais semblent néanmoins acceptables (voir la figure 4.3 pour une comparaison visuelle). Il est cependant difficile de généraliser cette conclusion à d'autres conditions (base d'apprentissage, nombre de vecteurs retenus, initialisation de l'algorithme, etc.).

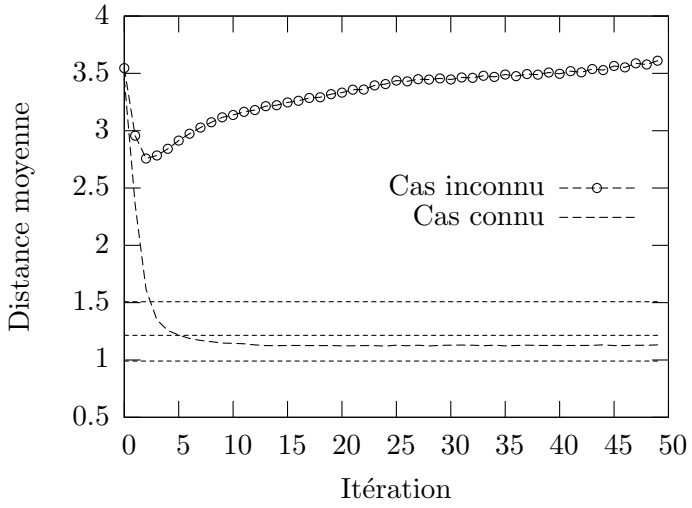


FIG. 4.2 – Distance moyenne à la vérité terrain en fonction de l'itération pour le cas connu et inconnu. Les valeurs maximales, minimales et moyennes des annotations manuelles de référence sont représentées par des lignes pointillées.

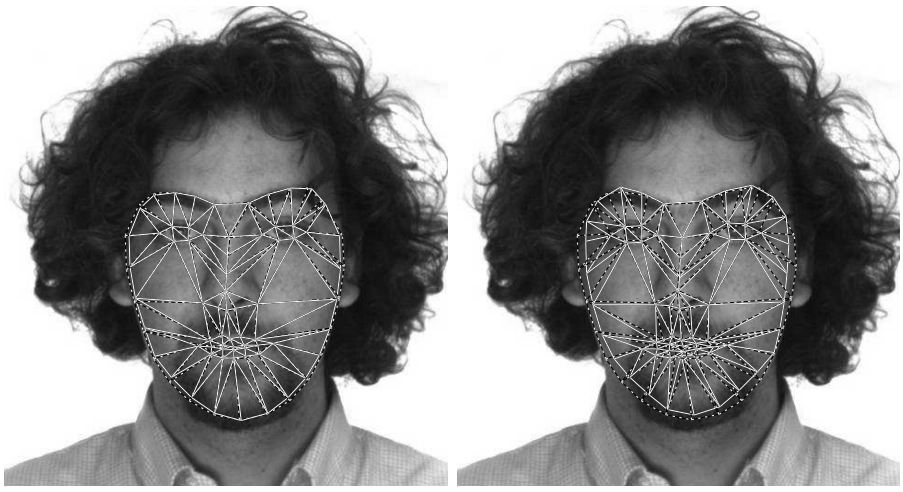


FIG. 4.3 – Résultats typiques de l'adaptation d'AAM lorsque le visage testé fait partie de la base (à gauche où $e(\mathbf{s}) = 1.16$) et lorsque le visage ne fait pas partie de la base d'apprentissage (à droite où $e(\mathbf{s}) = 3.46$). La forme vérité est tracée en pointillés.

4.3 Pouvoir de généralisation

La précision de l'algorithme d'adaptation d'AAM est maximale quand le visage observé est connu du modèle, c'est à dire quand la statistique de forme et de texture est capable de l'expliquer.

De manière à avoir une meilleure précision atteignable par l'algorithme, la logique voudrait qu'on ajoute plus d'exemples à la base d'apprentissage, ainsi un nouveau visage aurait plus de chance d'être proche (par interpolation linéaire) de l'ensemble des visages de la base.

Cependant, ajouter de nouveaux exemples à la base implique un nombre plus important de vecteurs de forme et de texture à retenir pour expliquer un même pourcentage de variance. Or, il se trouve que les performances de l'algorithme dépendent du nombre de vecteurs de forme et de texture retenus.

Il est nécessaire de distinguer le pouvoir de représentation du modèle, à savoir la capacité de la base des vecteurs de déformations et de la base des variations de texture à expliquer une nouvelle donnée et quelles difficultés présentent de telles bases pour l'algorithme d'adaptation.

Dans [Gross 05], les auteurs mènent une étude sur le pouvoir de généralisation des AAM. Trois ensemble de données sont distingués, chacun faisant varier idépendamment un des paramètres d'illumination, pose ou identité des visages. Chacun des ensembles contient 100 images, toutes annotées manuellement. Une première expérience consiste à construire un AAM sur chacun des trois ensembles de données, en retenant un nombre d'exemples croissant et à tester la reconstruction des visages d'un deuxième jeu indépendant de test. L'erreur de reconstruction est la distance entre la donnée de test et sa projection sur l'espace de formes ou d'apparences.

Il est ainsi possible de construire un modèle de forme qui généralise bien les différentes poses 3D avec 6 vecteurs de déformation. De la même façon, un modèle de forme généralisant l'identité peut être construit avec une quinzaine de vecteurs de déformations. Ces résultats ne sont cependant pas transposables à l'apparence. En effet, l'erreur de reconstruction de l'apparence pour un exemple hors de la base d'apprentissage est importante et l'ajout d'exemples à la base d'apprentissage ne fait diminuer l'erreur de reconstruction que de très peu. C'est pourquoi les auteurs concluent qu'il est difficile de construire un AAM qui soit capable de généraliser à de nouvelles données sans envisager une base d'apprentissage avec des milliers d'exemples.

La représentativité de l'apparence peut être cependant augmentée en segmentant le modèle de forme. Une approche de ce type est utilisée dans [Romdhani 04].

Concernant la difficulté d'adaptation, mesurée en terme de fréquence de convergence, construire un modèle en retenant de nombreux vecteurs de forme pose bien plus de difficultés que de retenir de nombreux vecteurs de texture.

Ceci s'explique par le fait que le modèle devient de plus en plus souple quand on lui ajoute des vecteurs de déformations possibles. Avec de nombreuses déformations possibles, la probabilité de tomber dans un minimum

local éloigné d'une solution satisfaisante augmente.

Bien que les tests aient été effectués par les auteurs de [Gross 05] pour déterminer la généralisation à l'identité, à la pose et aux conditions d'éclairage, la conclusion reste valable pour une généralisation à l'expression et à la pose.

Afin de remédier à ce problème, il est possible d'utiliser des modèles locaux spécialisés : soit en découpant le modèle de forme en plusieurs sous-parties localisées sur les composantes faciales, soit en partitionnant l'espace des formes en espaces de dimensions inférieures et en basculant d'un espace à l'autre en cours d'optimisation.

Nous utilisons une telle technique pour l'algorithme de suivi qui sera présenté dans le chapitre 5 : nous construisons un AAM rigide qui ne retient aucun vecteur de variation de forme et de texture. Les seules déformations possibles sont les déformations géométriques globales (similarités euclidiennes) : rotation (dans le plan), translation et mise à l'échelle. Bien que ce modèle soit très peu précis, il est néanmoins très robuste et permet de repositionner un modèle plus souple en cas d'échec.

4.4 Complexité, temps de calcul

Nous avons présenté précédemment l'algorithme *simultané*, qui optimise à la fois les paramètres de forme et les paramètres de texture. C'est l'algorithme qui donne les résultats les plus précis de la famille des algorithmes par composition inverse et donc celui qui peut être utilisé dans le cas le plus général.

Dans [Baker 03a], les auteurs annoncent une complexité d'une itération de l'algorithme simultané en $O((n+m)^2N + (n+m)^3)$, où n est le nombre de vecteurs de forme retenu, m le nombre de vecteurs de texture et N la résolution de la forme de référence $\mathbf{s}_0$.

Toujours dans le même article, les auteurs proposent alors plusieurs approximations de l'algorithme simultané. En particulier, la version appelée *project-out*, qui sera reprise pour application aux AAMs (dans [Matthews 03]).

L'idée est de suivre l'approche proposée dans [Hager 98]. La fonction d'erreur est séparée en une somme de deux erreurs calculées dans deux espaces vectoriels complémentaires : l'espace $\mathbf{T}$ engendré par l'ensemble des vecteurs de variations de texture $\mathbf{t}_i$ et son espace complémentaire $\mathbf{T}^\perp$.

$$\left\| \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s)) \right\|_{\mathbf{T}}^2 + \left\| \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s)) \right\|_{\mathbf{T}^\perp}^2$$

Dans le deuxième membre, l'erreur ne dépend plus de $\mathbf{p}^t$. On simplifie alors par :

$$\left\| \mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s)) \right\|_{\mathbf{T}}^2 + \|\mathbf{t}_0(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^s))\|_{\mathbf{T}^\perp}^2$$

Le calcul peut donc être décomposé en deux étapes : le calcul de $\mathbf{p}^s$ à partir du membre de droit, qui revient à l'application simple de l'algorithme sans variation de texture, en projetant l'image d'erreur dans l'espace complémentaire $\mathbf{T}^\perp$ et le calcul de $\mathbf{p}^t$ à partir du membre gauche, en ayant injecté au préalable le $\mathbf{p}^s$ calculé précédemment (le calcul est direct). On dit alors que les paramètres de texture $\mathbf{p}^t$ sont retrouvés par projection (*projected out*).

Lorsque l'optimisation est menée uniquement sur les paramètres de forme (comme c'est le cas lors de la première étape), les gradients et la matrice $\mathbf{H}$ peuvent être precalculés. La deuxième étape n'intervenant qu'après convergence de la première, l'algorithme résultant est très efficace. La complexité de cet algorithme est en $O(nN + m)$.

Cependant, dans la pratique, il s'avère que cet algorithme n'est précis que quand la texture est considérée comme variant très peu par rapport à l'estimation initiale. Lorsque la variation de texture est importante (par exemple quand la texture est initialisée à la texture moyenne d'une base de visages d'identités différentes), l'algorithme est très peu précis et généralement divergent, contrairement à l'algorithme simultané, moins efficace, mais beaucoup plus précis dans le cas général.

Il est à noter que le temps de calcul de l'algorithme simultané peut être réduit en utilisant certaines heuristiques, notamment en ne recalculant pas systématiquement les gradients à chaque itération.

Nous avons proposé une autre heuristique permettant de gagner en temps de calcul sur l'exécution de l'algorithme simultané [Mercier 06] : il s'agit de ne plus recourir à la matrice $\mathbf{H}$ à chaque itération.

En effet, dans l'algorithme simultané, l'étape de construction de cette matrice $\mathbf{H}$ est la plus coûteuse en temps de calcul.

L'équation 3.5 (p. 54) de mise à jour des paramètres :

$$[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t] = -\mathbf{H}^{-1} \sum_{\mathbf{x}} G(\mathbf{x})^T E(\mathbf{x})$$

devient alors :

$$[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t] = \mathbf{C} \sum_{\mathbf{x}} G(\mathbf{x})^T E(\mathbf{x})$$

où $\mathbf{C}$ est une matrice diagonale de coefficients pondérateurs :

$$\mathbf{C} = \begin{pmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_{n+m} \end{pmatrix}$$

Les coefficients du vecteur $\mathbf{C}$ sont alors obtenus par un raisonnement de type « descente de gradient » en chacun d'eux : d'une itération à l'autre, et pour chaque paramètre, tant que le paramètre évolue dans le même sens, il est encouragé dans ce sens et si le sens d'évolution est différent du sens précédent, il est freiné.

Le sens d'évolution est calculé par le signe de $\Delta\mathbf{p}_i(t)\Delta\mathbf{p}_i(t-1)$ (évolution donnée à l'itération t par l'équation de mise à jour du paramètre i comparée avec l'évolution calculée à l'itération précédente $(t-1)$).

pour $i = 1$ à $n + m$ **faire**
    si $\Delta\mathbf{p}_i(t)\Delta\mathbf{p}_i(t-1) > 0$ **alors**
        $c_i(t) \leftarrow c_i(t-1)\eta_{inc}$
    sinon
        $c_i(t) \leftarrow c_i(t-1)/\eta_{dec}$
    fin si
fin pour

Les paramètres η_{inc} et η_{dec} sont déterminés empiriquement.

La première itération est cependant identique à l'algorithme simultané, puisque la matrice $\mathbf{H}$ a pu être précalculée.

Une itération d'un tel algorithme est bien plus rapide qu'une itération de l'algorithme simultané. En revanche, il demande bien plus d'itérations pour arriver à la même précision que l'algorithme simultané. Le bilan, après tests, est que cet algorithme basé sur la régulation d'un vecteur de coefficients offre des performances équivalentes à l'algorithme simultané.

A titre indicatif, les méthodes présentées dans cette thèse ont été implantées via un ensemble de scripts pour le logiciel Gnu Octave. Avec cette implantation logicielle, l'algorithme simultané quand l'AAM considéré contient 13 vecteurs de forme et 23 vecteurs de texture (ce qui correspond à 95% de la variance de forme et de texture sur une base de 25 expressions d'un même visage) et que la résolution de $\mathbf{s}_0$ est de 48×48 pixels s'exécute en 0.75 seconde par itération. Dans les mêmes conditions, l'algorithme *project-out* s'exécute en 0.2 seconde par itération. L'algorithme simultané a un comportement bien plus convergent que le *project-out*, ce qui empêche une comparaison précise. En revanche, en considérant le gain de temps de calcul obtenu après traduction en un langage compilé et le très faible temps de calcul du *project-out* annoncé dans [Xiao 04], on peut envisager un suivi à la cadence vidéo pour le *project-out* (Xiao *et al.* annoncent un suivi à 230 images par seconde).

4.5 Résolution

Les performances des algorithmes d'adaptation d'AAM dépendent fortement de la résolution d'échantillonnage retenue.

Comme nous l'avons vu précédemment, la complexité algorithmique est fonction de cette résolution N.

De plus, la résolution d'échantillonnage indique la précision maximale atteignable par l'algorithme. Si le visage dans l'image d'entrée peut être décrit par une image de M pixels et que la résolution de $\mathbf{s}_0$ est de N pixels, alors les positions des points du modèle de forme peuvent être déterminées avec une précision maximale de ρ pixels, avec :

$$\rho = \frac{1}{2}\sqrt{\frac{M}{N}}$$

A titre indicatif, pour des visages décrits par environ 40 000 pixels (soit une fenêtre d'environ 200×200), nous utilisons généralement une forme $\mathbf{s}_0$ décrite par 64×64 pixels pour les résultats les plus précis et éventuellement par 48×48 pixels pour accélérer les calculs. Ceci correspond à une précision de l'ordre de 1.5 pixels dans le premier cas et d'un peu plus de 2 pixels dans le second cas. Compte tenu des défauts de mise au point et du bruit ajouté par la compression de la chaîne d'acquisition, ces précisions (maximales) sont raisonnables.

Il est à noter que dans le cas où la résolution de $\mathbf{s}_0$ est supérieure à la résolution de l'image fournie en entrée au système, l'algorithme présenté précédemment est sous-optimal. Ce contexte particulier a été étudié dans [Dedeoglu 06]. Il en résulte un algorithme qui prend explicitement en compte la différence de résolution et donne ainsi des résultats bien plus précis que la version originale.

4.6 Construction du modèle

Le nombre de points d'intérêt du modèle de forme n'influe pas directement sur les performances de l'algorithme. Comme nous l'avons vu en 4.4, la complexité algorithmique dépend directement de la résolution N de $\mathbf{s}_0$ et du nombre de vecteurs de forme et de texture retenus. Il se peut que le nombre de points influe légèrement sur les performances de l'algorithme de remplissage de texture (voir A.3), mais nous considérons cet effet négligeable.

Les points du modèle de forme permettent à certains phénomènes de variation non-linéaire de la texture d'être pris en compte en faisant que chaque modèle transformée vers la forme moyenne ait une texture dont les variations peuvent être expliquées par combinaison linéaire des vecteurs de la base des textures. Le but est de choisir un modèle de points tel que la texture contenue à l'intérieur de chacun des triangles pour toutes les images de la base d'apprentissage puisse être décrite avec le moins de vecteurs possibles.

Dans notre cas, nous avons utilisé deux types de modèles de forme : un modèle de forme à 68 points, identique à celui utilisé dans les articles de Baker & Matthews, pour comparaison, et un modèle à 35 points, plus compact, constitué de l'ensemble minimal de points à prendre en compte pour le suivi

d'expressions, selon notre estimation. Les triangles du modèle de forme ont été déterminés dans notre cas, par une triangulation de Delaunay.

Ainsi, pour un même ensemble d'apprentissage, il existe plusieurs modèles de forme possibles. Et parmi tous ces modèles de forme il en existe un qui, pour une représentativité maximale (l'erreur de reconstruction de toute la base d'apprentissage est minimale), minimise le nombre de vecteurs de forme et de texture. C'est une approche de ce type qui a été étudiée dans [Baker 04c] permettant d'envisager la construction automatique d'AAM.

4.7 Prise en compte des différents éclairages

Nous avons vu précédemment que les performances des algorithmes d'adaptation d'AAM dépendaient du fait que l'identité du visage fourni en entrée soit très proche (voire la même) que celle présente dans la base d'apprentissage.

De la même manière, la variation du type d'éclairage, entre celui de la base d'apprentissage et celui de l'image fournie en entrée influe grandement sur les performances de l'algorithme. En effet, l'image des résidus $E(\mathbf{x})$ est construite comme étant la différence des intensités lumineuses de l'image d'entrée et de l'estimation actuelle de la texture, construite d'après la base d'apprentissage.

De manière à prendre en compte un nouveau type d'éclairage, il est possible d'ajouter des exemples à la base d'apprentissage, tous éclairés de manière différente. Cependant, les variations d'éclairage étant un phénomène de nature non-linéaire, il serait nécessaire d'ajouter un nombre important d'exemples et de retenir un très grand nombre de vecteurs de variations de texture.

Une approche alternative consiste à modéliser le comportement de la lumière et à ajouter un paramètre d'illumination à l'ensemble des paramètres de texture à optimiser.

4.7.1 Modélisation de la lumière

Ainsi, une modélisation simple de la lumière consiste à considérer deux grandeurs : une valeur de biais et une valeur de gain de l'éclairage. Ces paramètres représentent la température de la lumière et la mise au point photogrammétrique du capteur.

Si on ajoute le vecteur $\mathbf{t}_0$ (qui est la texture moyenne) aux vecteurs de variations de texture, alors le coefficient qui lui est associé est une approximation du gain. De même, en ajoutant le vecteur unitaire, rempli de 1, alors le coefficient qui lui est associé représente le biais.

Cette heuristique, que nous avons utilisée, permet donc de prendre en compte des différences de gain et de biais au niveau de l'éclairage (qui peuvent survenir par exemple lors d'une vidéo, lorsque la caméra adapte automatiquement ces valeurs).

Il est à noter tout de même, qu'il n'est pas possible d'utiliser directement l'algorithme *project-out* dans ce cas (voir [Baker 03a] pour une justification

détaillée).

On peut imaginer de même que toute variation linéaire d'éclairage peut être prise en compte de cette façon. Par exemple, en ajoutant un vecteur contenant des 1 sur le côté droit du visage et 0 sur le côté gauche, l'algorithme sera plus en mesure de prendre en compte des éclairages de côté.

La généralisation à la prise en compte d'un éclairage d'orientation quelconque n'est pas aisée. En effet, il serait nécessaire alors d'avoir un vecteur de variation de texture par orientation de l'éclairage, résultant en un nombre important de vecteurs à manipuler.

4.7.2 Modélisation de la couleur

Une autre approche consiste à trouver une représentation de l'image qui soit équivalente quelque soit les types d'éclairage de la scène.

L'idée est alors de filtrer les images de manière à décorréler la couleur de la peau et la lumière qui l'éclaire. De nombreuses transformées de la couleur existent, ayant chacune certains invariants. On peut trouver un inventaire des différentes transformations colorimétriques dans [Gevers 97] avec, pour chaque transformée, le type de propriété lumineuse à laquelle elle est invariante.

Certaines de ces composantes peuvent conserver plusieurs composantes pour représenter une couleur indépendante de l'éclairage. Pour pouvoir utiliser cette représentation avec les AAM, il est donc nécessaire au préalable d'étendre les algorithmes d'adaptation d'AAM à plusieurs composantes.

Dans la pratique, nous avons appliqué les algorithmes d'adaptation d'AAM sur des images dont l'éclairage ne diffère de celui de l'ensemble d'apprentissage que par une différence de gain ou de biais, considérant qu'il n'existait pas encore de technique satisfaisante de prise en compte des différents types d'éclairage dans le cas général.

Troisième partie

Applications au contexte de la langue des signes

Chapitre 5

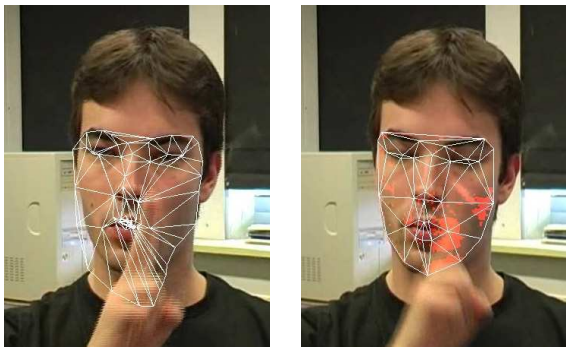
Prise en compte des occultations

En LSF, de nombreux signes sont produits près ou dans l'axe du visage du signeur. De plus il est fréquent que le visage du signeur soit en rotation (particulièrement lors de structures grammaticales appelées transferts personnels). Ceci implique que du point de vue de l'interlocuteur (ici remplacé par le système de captation vidéo) le visage n'est généralement vu que partiellement.

L'interlocuteur met ainsi en place un mécanisme robuste de compréhension : le sens porté par le visage du signeur (valeur aspectuelle ou modale) est capté bien que les indices visuels faciaux ne soient que partiellement visibles. Ceci parce que le système humain de vision permet une certaine interpolation de l'information (notamment dans le temps) et parce que les signes occultant totalement le visage pendant longtemps sont rares et signifient que le visage est caché (*un masque, se laver le visage, flouter le visage, avoir une vision floue*). Dans ce cas, il n'y a pas de perte d'information due au masquage du visage.

Sans traitement particulier, un algorithme d'adaptation d'AAM considère qu'une instance du modèle déformable appris est présente dans l'image. En cas d'occultation partielle, une partie du modèle sera déformée pour la prendre le plus possible en compte alors qu'elle n'existe pas dans la statistique : pour expliquer les occultations, des coefficients de déformation très forts seront appliqués au modèle, amenant à un modèle ne représentant plus un visage. Au lieu d'ignorer cette partie du modèle, l'algorithme va au contraire lui donner un poids important pour arriver à expliquer l'observation.

L'algorithme d'adaptation doit donc être capable de déformer le modèle en ne prenant en compte que les pixels de l'image qui ne sont pas occultés.



(a) (b)

FIG. 5.1 – Adaptation d'un modèle déformable lors de la présence d'occultation (a) sans prise en compte de données aberrantes, (b) avec prise en compte des données aberrantes par l'utilisation d'une carte de confiance.

5.1 Variante robuste

Du point de vue de l'algorithme d'adaptation, une occultation peut être vue comme une partie de l'image à ne pas prendre en compte ou peu. En particulier, on peut considérer que l'image des résidus est pondérée en chacun de ses pixels. Le poids associé à chaque pixel correspond au degré de confiance et, dans le cas de l'occultation, au degré de non-occultation.

Baker et Matthews définissent une variante de leur algorithme utilisant une pondération en chacun des pixels [Baker 03b]. La nouvelle fonction à minimiser est :

$$\sum_{\mathbf{x}} Q(\mathbf{x}) \left[\mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(x) - I(W(\mathbf{x}; \mathbf{p}^s)) \right]^2$$

où $Q(\mathbf{x})$ est une carte, pondérant l'influence de chacun des pixels $\mathbf{x}$.

Les détails de calcul permettant la dérivation de la fonction d'erreur sont présentés dans [Baker 03b] et repris en A.1§

$$\mathbf{H} = \sum_{\mathbf{x}} Q(\mathbf{x}) G(\mathbf{x})^T G(\mathbf{x})$$

et le calcul de l'évolution des paramètres :

$$[\Delta \mathbf{p}^s, \Delta \mathbf{p}^t] = -\mathbf{H}^{-1} \sum_{\mathbf{x}} Q(\mathbf{x}) G^T(\mathbf{x}) E(\mathbf{x}) \tag{5.1}$$

Les étapes de l'algorithme, appelé alors « simultané pondéré », sont représentées sur la figure 5.2.

La carte $Q(\mathbf{x})$ peut être calculée à partie de l'image des résidus $E(\mathbf{x})$. De nombreuses fonctions dites robustes, qui croissent moins vite que l'identité à partir d'un certain seuil existent dans la littérature.

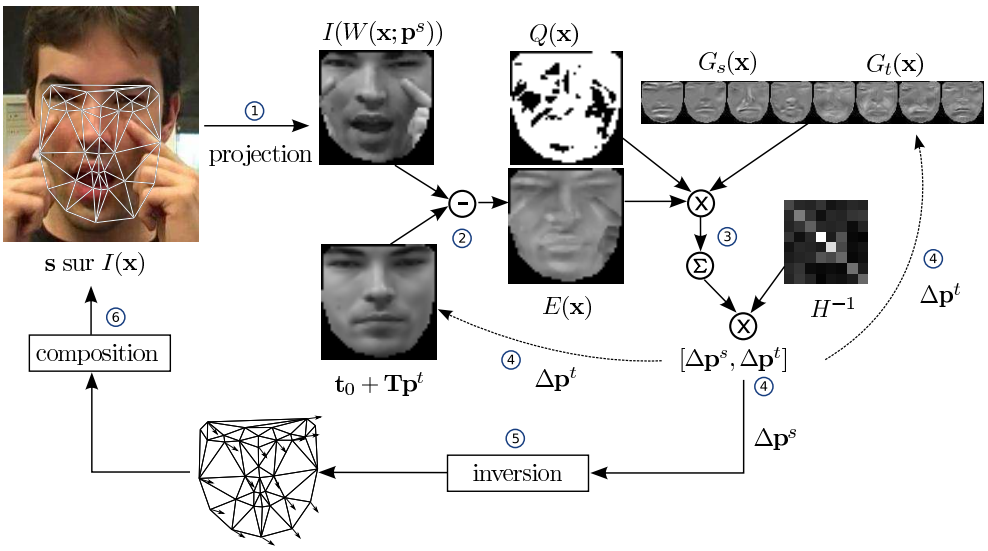


FIG. 5.2 – Étapes d'une itération de l'algorithme simultané pondéré.

Les fonctions robustes dépendent toutes d'un paramètre de seuil σ à partir duquel la fonction commence à changer de comportement. C'est généralement à partir de ce seuil qu'une importance moindre est donnée à un pixel particulier, supposant que plus l'erreur est importante, plus le pixel considéré a de chance d'être considéré comme « aberrant ».

La détermination d'un seuil σ capable de distinguer un pixel aberrant d'un autre est une tâche difficile en particulier pour qu'il soit suffisamment discriminant quelque soit la situation.

Objectifs Les objectifs de la prise en compte des occultations sont :

- la détection des zones occultées du visage afin que l'algorithme d'adaptation d'AAM arrive à converger vers une position où le modèle est bien adapté sur les zones non occultées ;
- la segmentation correcte des zones occultées du visage ; en particulier les zones non-occultées ne doivent pas être considérées comme occultées ;
- le suivi correct d'une séquence vidéo comprenant des occultations : l'adaptation du modèle déformable sur une image non-occultée ne doit pas être perturbée par une divergence sur une image occultée qui précède.

De plus, ces objectifs doivent être atteints le plus possible de manière automatique.

La carte de confiance $Q(\mathbf{x})$ utilisée dans la variante pondérée de l'algorithme d'adaptation d'AAM doit être aussi proche que possible de la carte des occultations. Si $M(\mathbf{x})$ est la carte des occultations réelles *i.e.*, une image binaire de même dimensions que I où chaque pixel vaut 1 si $I(\mathbf{x})$ est occulté et 0 sinon, alors la carte de confiance idéale à utiliser à chaque itération est $\mathbf{1} - M(W(\mathbf{x};\mathbf{p}^s)), \forall \mathbf{x} \in \mathbf{s}_0$.

Le problème est de calculer la meilleure carte de confiance sans connaissance sur la localisation des occultations réelles. Nous proposons pour ce faire de modéliser le comportement de l'image des résidus dans le cas non-occulté et de détecter les occultations comme étant ce qui n'est pas bien expliqué par le modèle, suivant l'approche présentée dans [Theobald 06].

5.1.1 Modèles paramétriques des résidus

Nous utilisons des modèles paramétriques de l'image des résidus.

Nous proposons de tester différents calculs de la carte de confiance :

$$Q_1(\mathbf{x}) = \begin{cases} 1 & \text{si } \min(\mathbf{x}) \leq E(\mathbf{x}) \leq \max(\mathbf{x}) \\ 0 & \text{sinon} \end{cases}$$

$$Q_2(\mathbf{x}) = \frac{1}{\sigma(\mathbf{x})\sqrt{2\pi}} e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

$$Q_3(\mathbf{x}) = \begin{cases} 1 & \text{si } |E(\mathbf{x})| \leq 3\sigma(\mathbf{x}) \\ 0 & \text{sinon} \end{cases}$$

$$Q_4(\mathbf{x}) = \begin{cases} 1 & \text{si } |E(\mathbf{x})| \leq 4\sigma(\mathbf{x}) \\ 0 & \text{sinon} \end{cases}$$

$$Q_5(\mathbf{x}) = e^{\left(-\frac{E(\mathbf{x})^2}{2\sigma(\mathbf{x})^2}\right)}$$

Dans les fonctions ci-dessus, $\min(\mathbf{x})$ est la valeur minimale du pixel $\mathbf{x}$ sur toutes les images des résidus, $\max(\mathbf{x})$ est la valeur maximale et $\sigma^2(\mathbf{x})$ la variance.

L'apprentissage des paramètres pourrait être effectué sur un ensemble quelconque d'images des résidus généré quand l'algorithme d'adaptation est utilisé sur des images non-occultées.

Cependant, une image des résidus générée quand le modèle est éloigné de la solution est très différente d'une image générée quand le modèle est proche de la solution (voir à ce propos la figure 5.3).

Ainsi les paramètres des modèles des résidus dépendent de la distance du modèle à la solution : ils doivent être permissifs quand le modèle est éloigné de la solution et stricts quand le modèle est proche de la solution.

Ensemble d'apprentissages partitionnés

De manière à expliciter le lien entre les paramètres et la distance à la solution, nous avons procédé au test suivant.

Un ensemble d'images des résidus est généré : l'algorithme d'adaptation d'AAM (non-pondéré) est lancé à partir des formes optimales perturbées pendant 15 itérations jusqu'à convergence. Pour initialiser l'AAM, les coordonnées

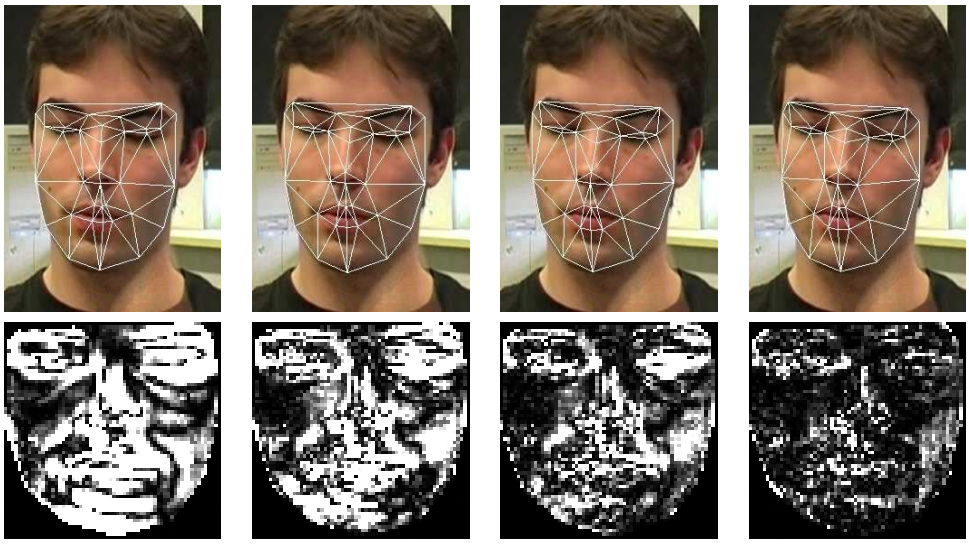


FIG. 5.3 – Exemple d'évolution de l'image des résidus $E(\mathbf{x})^2$ (en bas) suivant l'évolution de la position du modèle de forme sur des images non-occultées (en haut).

de chaque point de la forme optimale sont perturbées par un bruit gaussien ayant 10 variances différentes (entre 5 et 30). L'algorithme est lancé 4 fois sur 25 images qui font partie de l'ensemble d'apprentissage de l'AAM. La distance à la solution, calculée par la distance euclidienne moyenne du modèle de forme au modèle de forme optimal, et l'image des résidus sont stockées à chaque itération.

Au lieu de calculer les paramètres ($\min(\mathbf{x})$, $\max(\mathbf{x})$ et $\sigma(\mathbf{x})$) sur toutes les images des résidus, nous formons 15 partitions en regroupant les images des résidus par rapport à leur distance à la solution. Chaque partition P_i contient 210 images des résidus et peut être caractérisée par sa distance minimale d_i^- et maximale d_i^+ à la solution. Les paramètres sont alors appris, pour chaque pixel $\mathbf{x}$, sur les résidus de chaque partition.

Sur la figure 5.4 sont représentés les écart-types $\sigma(\mathbf{x})$ appris sur chacune des partitions. Pour des raisons de visualisation, seul l'écart-type moyen σ, calculé en moyennant sur l'ensemble des pixels $\mathbf{x}$, est affiché.

5.1.2 Approximation des paramètres

Quand l'algorithme d'adaptation est lancé sur une image de test, la distance du modèle à la solution est difficile à estimer. En effet, la seule information disponible est l'image des résidus qui peut donner une estimation de la distance à la solution seulement dans le cas non-occulté. Une telle information n'est pas fiable dans le cas occulté, puisque les résidus reflètent aussi bien les erreurs de mauvais placement que les erreurs dues aux occultations.

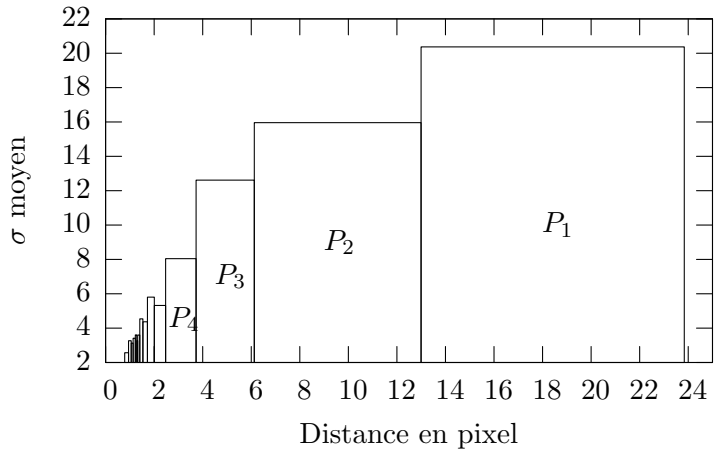


FIG. 5.4 – Écart-type moyen appris pour chaque partition.

Cependant, nous supposons que nous pouvons considérer le numéro d'itération de l'algorithme pour sélectionner la partition appropriée, en particulier si la distance du modèle à la solution dans le cas occulté est plus petite que la distance maximale utilisée pour regrouper les résidus de la première partition.

Pour valider cette hypothèse, nous avons procédé au test suivant. En utilisant les variances calculées sur chacune des 15 partitions, nous avons testé l'algorithme pondéré lancé pendant 20 itérations à partir de positions optimales perturbées par un bruit gaussien (avec une variance de 20) sur des images occultées (25% de l'image est couverte de blocs de 8×8 pixels d'intensité aléatoire). Il est à noter que les perturbations de la forme sont ici moins importantes que celles utilisées lors de la construction des partitions. Parmi toutes les fonctions $Q_i(\mathbf{x})$, nous utilisons $Q_3(\mathbf{x})$ pour calculer la carte de confiance à chaque itération. Un autre choix aurait pu être fait, puisque nous sommes seulement intéressés par la manière de calculer son paramètre, non par sa performance. Différentes manières de sélectionner la variance à chaque itération sont testées :

- S_{real} : sélection à partir de la partition P_i où la distance réelle à la solution d_{model} est bornée par l'intervalle de distances de P_i : $[d_i^-, d_i^+]$; pour comparaison ;
- S_{it} : sélection à partir de P_i où i est l'itération actuelle (et $i = 15$ pour les itérations 15 à 20) ;
- S_f : sélection à partir de P_1 ;
- S_m : sélection à partir de P_7 ;
- S_l : sélection à partir de P_{15}.

Les résultats sur la figure 5.5 montrent clairement que le meilleur choix pour le calcul du paramètre de l'image des résidus est S_{real}. Ce calcul n'est pas utilisable en pratique (la forme optimale n'est pas connue *a priori*), mais nous pouvons raisonnablement nous rabattre sur l'approximation S_{it}. Pour

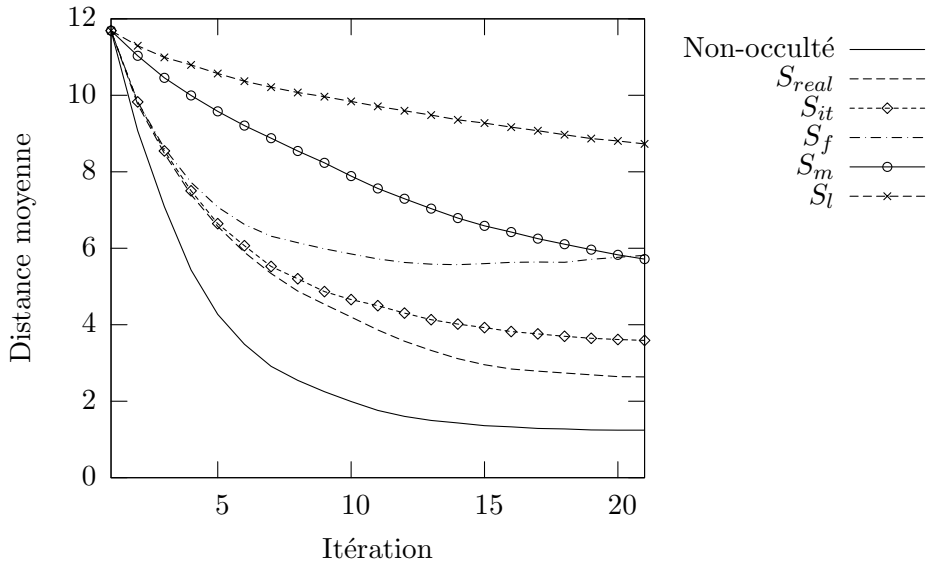


FIG. 5.5 – Comportement moyen de l'algorithme d'adaptation pour le cas non-occulté de référence et pour différents calculs de la variance dans le cas occulté.

comparaison, les résultats sont aussi donnés pour le cas non-occulté et pour des variances fixées (S_f, S_m et S_l). Toutes les variances fixes donnent de mauvais résultats comparé à S_{real} ou S_{it}.

5.1.3 Choix du modèle paramétrique

Avec les résultats précédents, nous pouvons tester quelle est la meilleure façon de calculer la carte de confiance utilisée à chaque itération.

Pour ce faire, nous procédons au test suivant : la version robuste de l'algorithme d'adaptation est lancée sur les images de la base d'apprentissage de l'AAM, couvertes avec un pourcentage variable d'occultations, depuis des formes perturbées par une gaussienne (nous utilisons une variance de 20 pour chaque coordonnée). Nous testons chacune des fonctions de calcul Q_i de la carte de confiance.

La fréquence de convergence est déterminée en calculant le nombre d'adaptations qui convergent vers une forme ayant une distance moyenne à la forme optimale inférieure à 2 pixels.

Les résultats sont résumés en FIG. 5.6. La fonction Q_4 montre clairement les meilleurs résultats. Toutes les autres fonctions donnent des résultats moindres, excepté pour la fonction Q_1 qui semble être un bon détecteur dans le cas d'un faible taux d'occultations et un très mauvais dans le cas d'un fort taux d'occultations. La fonction Q_1 repose sur le calcul de valeurs minimales et maximales, qui sont des mesures très bruitées, comparées à la variance. C'est pourquoi le comportement de Q_1 n'est pas toujours fiable.

Il apparaît aussi que toutes les fonctions amènent à divergence lorsque le taux d'occultations est supérieur à 50%.

5.2 Stratégie de suivi robuste

Notre objectif est que l'algorithme de suivi prenne le plus possible en compte les occultations. Cependant, sur certaines images, les occultations sont trop importantes pour s'attendre à une bonne adaptation du modèle, parce que très peu d'informations fiables existent. Dans une telle situation, l'algorithme d'adaptation a généralement un comportement divergent résultant en une forme qui serait une mauvaise initialisation si elle était utilisée directement dans l'image suivante.

C'est pourquoi nous proposons d'utiliser une mesure de divergence et un AAM rigide pour initialiser le modèle.

Le but est d'éviter les mauvaises configurations du modèle de forme, de manière à ne pas perturber le processus d'adaptation sur les images suivantes. Nous détectons ces mauvaises configurations comme étant celles mal expliquées par la statistique. Dans ce but, nous comparons les paramètres de forme $\mathbf{p}^s$ à leurs écarts types σ_i, qui ont été préalablement appris depuis la base d'apprentissage des formes. Comme présentée en 4.1, la divergence est décidée si :

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{p}_i^s|}{\sigma_i} > \rho_1 \text{ ou } \max_{i=1,\ldots,n}\left\{\frac{|\mathbf{p}_i^s|}{\sigma_i}\right\} > \rho_2$$

Les seuils ρ_1 et ρ_2 sont déterminés empiriquement et peuvent être hauts (nous choisissons ici $\rho_1 = 2.5$ et $\rho_2 = 7.0$). Les seuils sont testés seulement après dix itérations, puisque les déformations du modèle des premières itérations peuvent amener à convergence.

Sur chaque image, si la convergence est détectée, la configuration finale est stockée et elle sert comme initialisation pour l'image suivante.

Si une divergence est détectée, un modèle robuste est utilisé pour initialiser l'image suivante : un AAM construit en ne retenant que les vecteurs de déformation géométrique. Il s'agit d'un modèle représenté par la forme moyenne (et la texture moyenne) qui ne peut varier qu'en facteur d'échelle, rotation (dans le plan) et position mais pas en déformations faciales. Un tel modèle donne une estimation de la forme du visage qui peut être utilisée comme initialisation pour l'AAM non-rigide. Il empêche le modèle non-rigide d'être attiré par des minima locaux (présents dans l'arrière-plan de l'image par exemple). L'adaptation du modèle rigide utilise aussi une carte de confiance pour traiter les occultations. Cependant, celle calculée pour l'AAM non-rigide est trop stricte pour le modèle rigide, c'est pourquoi nous utilisons une carte plus permissive (dont la variance est calculée sur la deuxième partition par exemple).

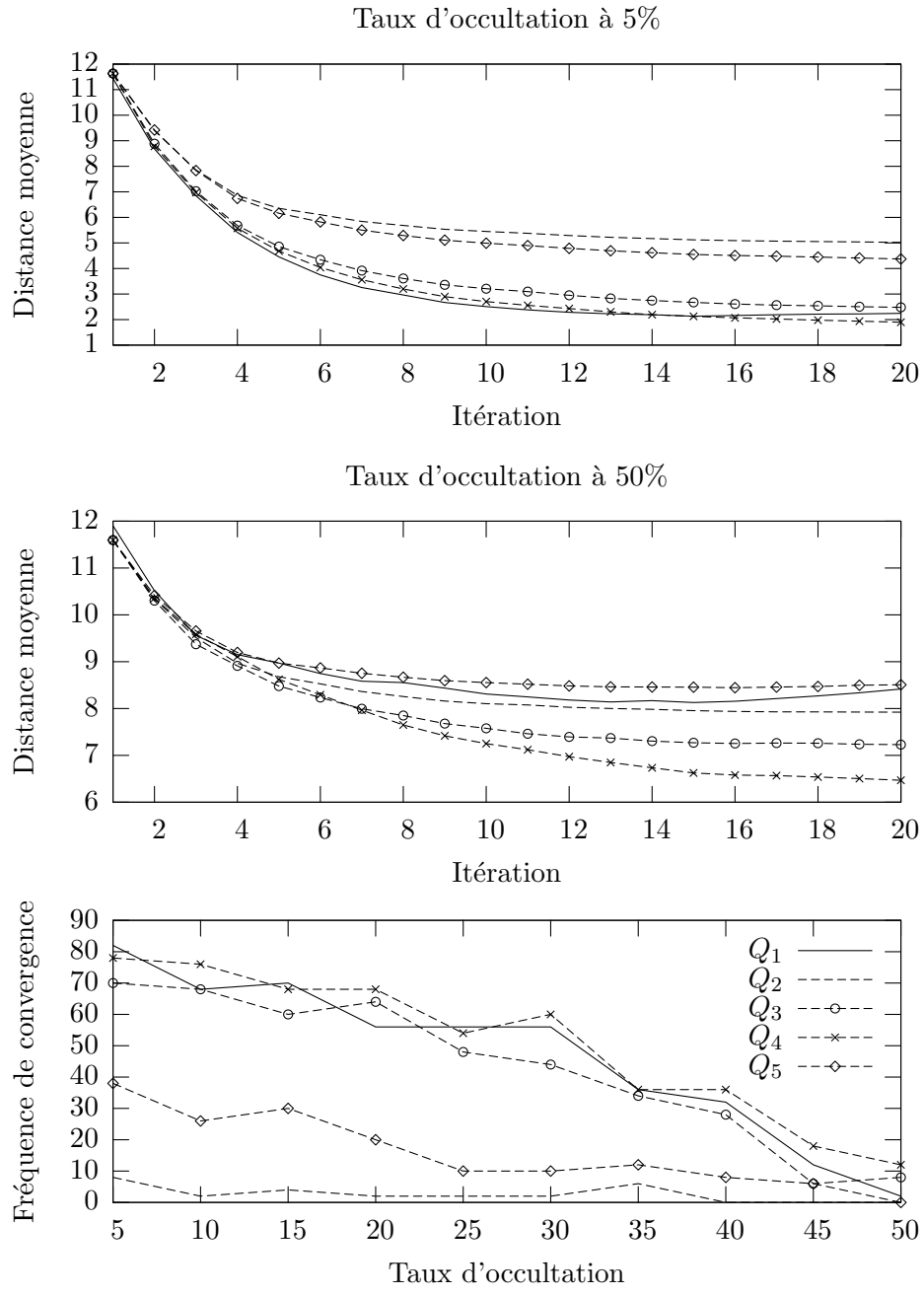


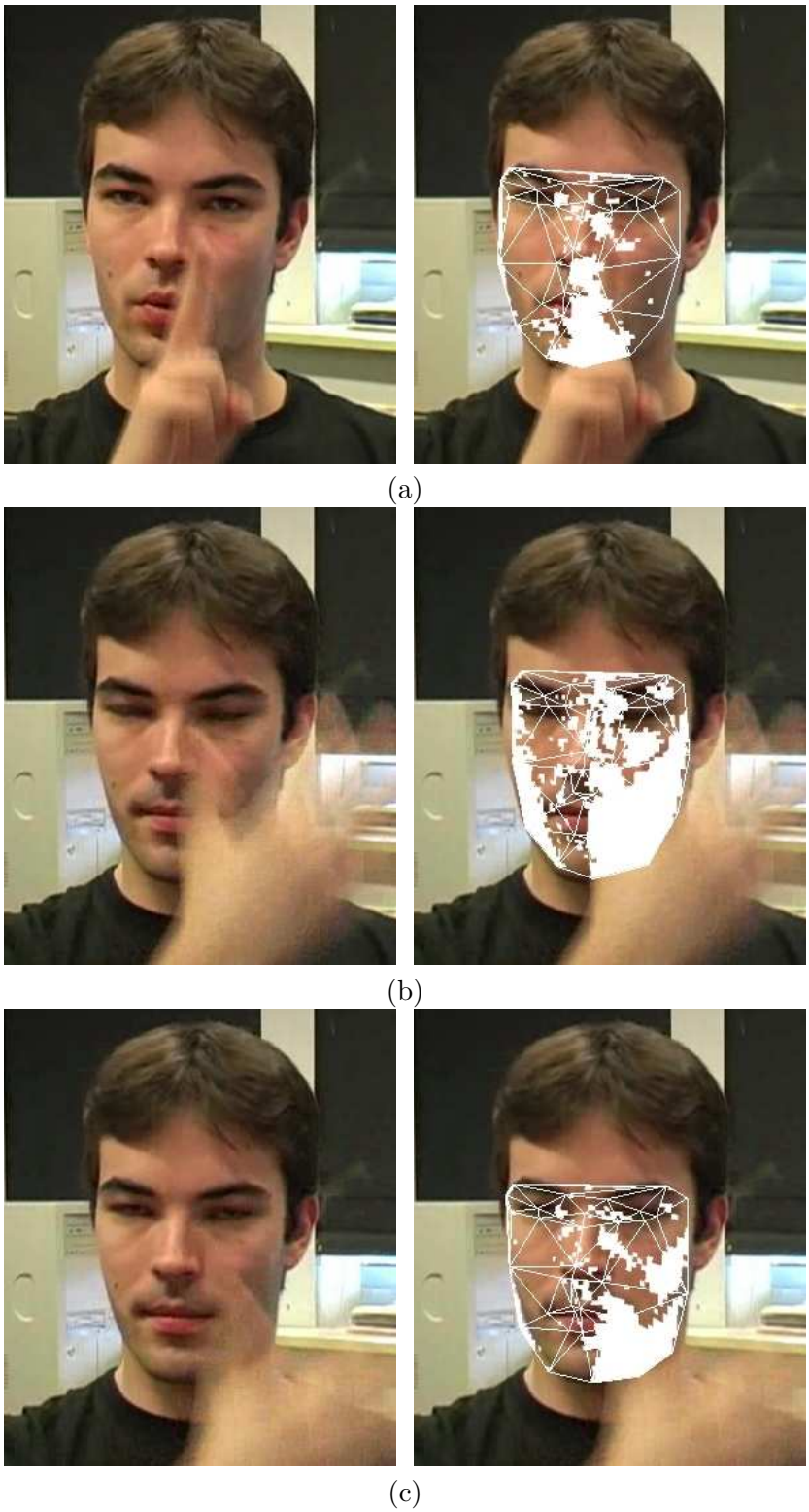
FIG. 5.6 – Caractérisation des calculs de la carte de confiance. La distance moyenne à la solution par itération pour 5% et 50% d'occultations (courbes du haut) et fréquence de convergence (courbe du bas).

L'adaptation du modèle rigide est lancée pendant 5 itérations à partir de la dernière configuration obtenue après convergence. Le modèle non rigide est ensuite lancé depuis la position résultante.

Nous avons testé cet algorithme de suivi sur une séquence vidéo d'environ 500 images où des signes viennent fréquemment occulter le visage du locuteur.

Quelques résultats typiques sont représentés en figure 5.7. Chaque point de la forme est affiché avec un niveau de gris calculé à partir de la carte de confiance. Comparé à un suivi naïf, l'AAM converge ici toujours quand il est lancé sur des images non-occultées.

Nous avons ainsi une méthode de suivi réaliste car suffisamment robuste pour prendre en compte les occultations inévitables rencontrées dans un corpus en langue des signes.



(a)

(b)

(c)

FIG. 5.7 – Résultats de suivi sur séquence vidéo. A gauche : image extraite de la séquence vidéo. A droite : maillage du modèle de forme et carte des occultations détectées. (a) : Exemple d'une bonne détection d'occultations. (b) et (c) : Divergence sur une image et convergence sur l'image suivante.

Chapitre 6

Applications

Nous décrivons dans ce chapitre deux applications de l'algorithme de suivi de déformations faciales présenté précédemment : il s'agit de la description d'expressions et de l'anonymisation. Nous avons choisi ces deux applications parce qu'elles sont pertinentes dans le cas de la langue des signes.

6.1 Description des expressions

L'algorithme présenté précédemment permet de suivre les déformations faciales au cours d'une vidéo. Le résultat du traitement est un ensemble de positions 2D d'un certain nombre de points de référence. Nous présentons dans cette section comment extraire une information sur les expressions à partir de ces informations.

Dans les domaines où l'analyse des expressions faciales est importante, on aimerait obtenir une information sur les muscles faciaux activés et sur leur intensité d'activation pour chaque image d'une séquence vidéo.

L'unité atomique de description utilisée ici est proche de l'*action unit* du système FACS. Elle est cependant davantage basée sur une décomposition visuelle plutôt qu'une décomposition musculaire. Une expression peut être décrite par une combinaison d'*action units* dans FACS. Nous distinguons deux types de combinaisons :

- la coarticulation, indiquant que plusieurs déformations affectent une même composante faciale ; c'est le cas par exemple des AU 1 + 2 + 5 qui ont lieu dans l'expression de peur et qui affectent les sourcils ;
- la cooccurrence, indiquant que plusieurs déformations surviennent chacune sur plusieurs composantes faciales différentes ; les mouvements du haut du visage sont par exemple généralement indépendants de ceux du bas du visage.

Les modèles à apparence active sont construits avec la supposition qu'un visage quelconque peut être représenté par une combinaison linéaire de formes et de textures.

L'objectif de la description des expressions est de fournir à chaque instant, un vecteur de paramètres $\mathbf{b}$ codant l'intensité de chacune des déformations unitaires. Une valeur nulle indique que la déformation unitaire n'est pas activée et une valeur à 1 indique une activation avec une intensité maximale.

L'approche naïve pour la description des expressions consiste à supposer que l'on dispose *a priori* d'une description exhaustive de chacune des déformations unitaires et de leurs combinaisons pour la personne dont ont suit les déformations. Il s'agirait d'un ensemble de séquences vidéos où la personne déforme son visage pour chacune des unités retenues en partant de l'expression neutre et jusqu'à son maximum d'intensité. Une forme et une texture seraient associées à chacune des images de chaque vidéo (soit par annotation manuelle soit comme résultat d'un algorithme d'adaptation d'AAM). Serait de plus associée une intensité d'activation à chaque image (avec la première à 0, la dernière à 1 et le reste obtenu par interpolation). La description consisterait alors à comparer la forme et texture extraite de la vidéo de test à celles stockées lors de l'apprentissage, par une méthode de classification, et d'en retourner l'intensité d'activation correspondante.

Cette approche pose un problème évident de stockage et nécessite de plus une base d'apprentissage très difficile à acquérir.

Nous proposons alors de considérer que l'intensité d'une expression peut être décrite par une interpolation linéaire entre l'expression neutre et l'expression d'intensité maximale. Autrement dit, qu'un visage de forme $\mathbf{s}$ et de texture $\mathbf{t}$ peut être décrit par :

$$\mathbf{s} = \mathbf{s}_n + \alpha(\mathbf{s}_m - \mathbf{s}_n)$$

et

$$\mathbf{t} = \mathbf{t}_n + \alpha(\mathbf{t}_m - \mathbf{t}_n)$$

avec $\alpha \in [0, 1]$ représentant l'intensité d'activation de l'expression, $\mathbf{s}_n$ et $\mathbf{t}_n$ la forme et la texture de l'expression neutre et $\mathbf{s}_m$ et $\mathbf{t}_m$ la forme et la texture de l'expression à son intensité maximale.

Ainsi, connaissant $\mathbf{s}_n$ et $\mathbf{s}_m$, l'intensité d'activation α de l'expression sur une forme $\mathbf{s}$ est obtenue par :

$$\alpha = (\mathbf{s}_m - \mathbf{s}_n)^+(\mathbf{s} - \mathbf{s}_n)$$

ou à partir de la texture :

$$\alpha = (\mathbf{t}_m - \mathbf{t}_n)^+(\mathbf{t} - \mathbf{t}_n)$$

où $.^+$ désigne la pseudo-inverse.

Cette modélisation est évidemment une approximation. Afin d'en visualiser la qualité, nous avons conduit le test suivant : sur une séquence vidéo représentant l'activation d'une déformation faciale entre l'intensité nulle et sa valeur maximale, un AAM ayant un modèle de forme à 35 points a été adapté

sur chaque image (il s'agit d'un sourire, extrait de la base de visages MMI [Pantic 05]). Cette déformation faciale peut être considérée comme composée de plusieurs unités d'action FACS (AU6 et AU12 en particulier). Nous avons considéré, dans un premier temps, cette combinaison comme une déformation unitaire.

Dans un premier temps, nous avons représenté sur un même graphique (figure 6.1) toutes les formes de la séquence (précédemment alignées par une analyse de Procrustes) ainsi que l'approximation linéaire faite. En rappelant que les points du contour du visage sont souvent localisés avec peu de précision, il est clair que l'approximation linéaire est une approximation réaliste dans ce cas.

La même comparaison visuelle étant difficile à faire pour les textures, nous présentons l'ensemble des textures de la séquence d'images ainsi que leur reconstruction par interpolation linéaire sur la figure 6.2 (les textures sont projetées sur une image de résolution 48×48).

De plus, sur chaque image a été calculée une erreur de reconstruction, (qui est très proche de la somme des distances que minimise la pseudo-inverse) :

$$e_{\mathbf{s}} = \frac{||\mathbf{s} - ((\mathbf{s}_m - \mathbf{s}_n)(\mathbf{s}_m - \mathbf{s}_n)^+(\mathbf{s} - \mathbf{s}_n) + \mathbf{s}_n)||}{||\mathbf{s}||}$$

et une erreur de reconstruction de la texture de la même manière. Cette expression permet de représenter l'erreur de reconstruction comme étant un pourcentage de la norme de la forme $\mathbf{s}$.

Pour chaque image de la séquence, le paramètre d'intensité α a été calculé et est représenté sur la figure 6.3. Les erreurs de reconstruction correspondantes sont représentées sur la figure 6.4.

Il apparaît que la modélisation par interpolation linéaire donne de bons résultats dans le cas du sourire. L'erreur de reconstruction maximale d'environ 11% est obtenue sur l'image numéro 20 (voir pour illustration les différences de texture sur la figure 6.2).

6.1.1 Cooccurrence d'expressions

Lorsque l'on désire décrire les intensités d'un ensemble de déformations unitaires, il est nécessaire d'avoir recours à une base d'apprentissage contenant le visage neutre en expression et le visage affichant chacune des déformations faciales à leur maximum d'intensité. On soustrait l'expression neutre de chacune des formes et textures d'intensité maximale et on concatène le résultat dans une matrice $\mathbf{B}$:

$$\mathbf{B}_{\mathbf{s}} = [(\mathbf{s}_{m,1} - \mathbf{s}_n), \quad \ldots, \quad (\mathbf{s}_{m,N} - \mathbf{s}_n)]$$

où $\mathbf{s}_{m,i}$ représente la forme correspondant à l'expression i à son intensité maximale.

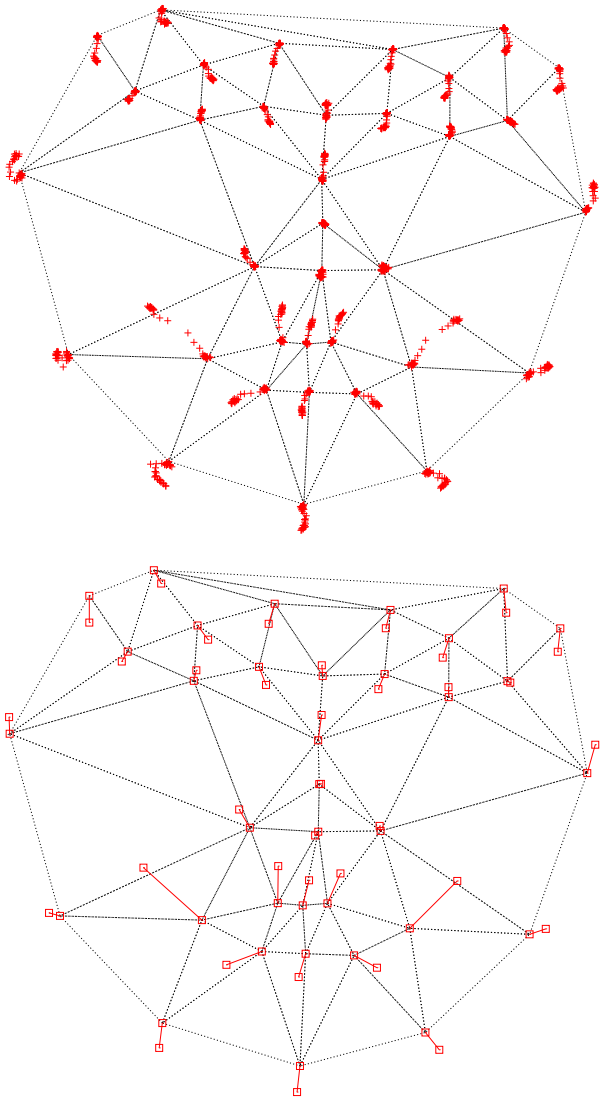


FIG. 6.1 – Ensemble des formes de la séquence « sourire » (en haut) et approximation linéaire utilisée (en bas).

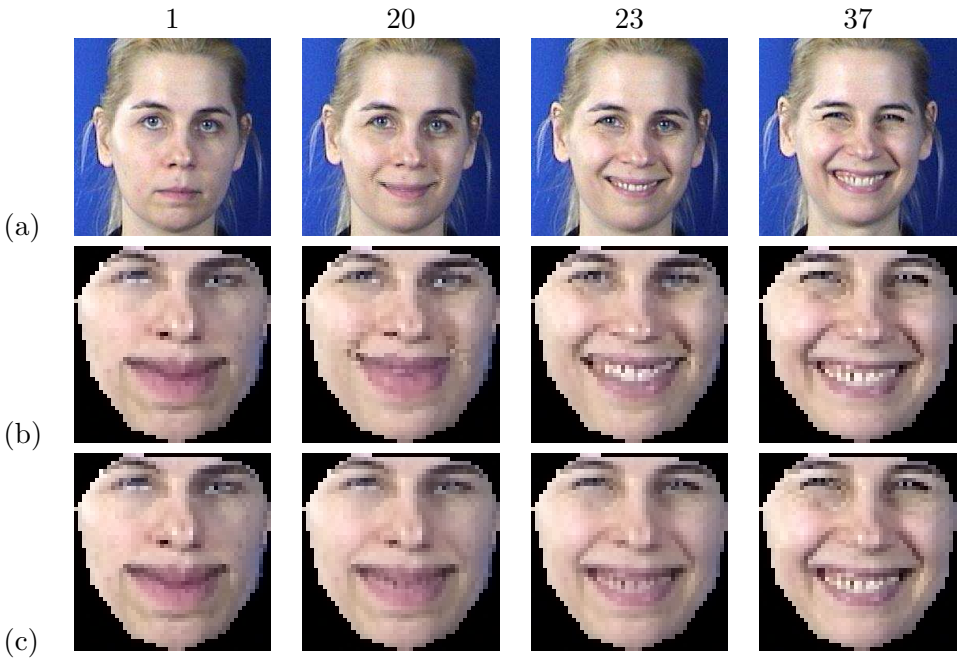


FIG. 6.2 – (a) Images extraites de la séquence « sourire ». (b) Texture correspondante. (c) Reconstruction de la texture.

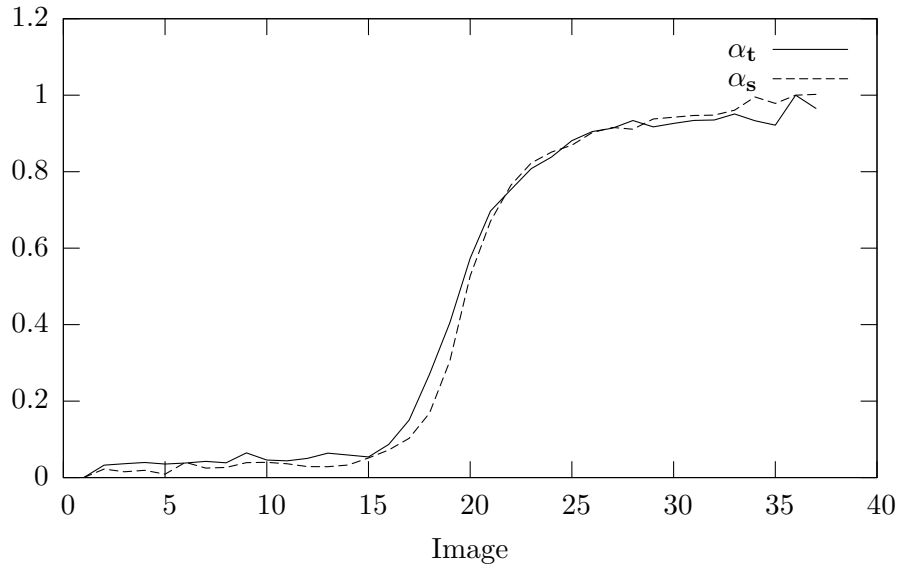


FIG. 6.3 – Paramètre d'intensité de l'expression de sourire au cours de la séquence.

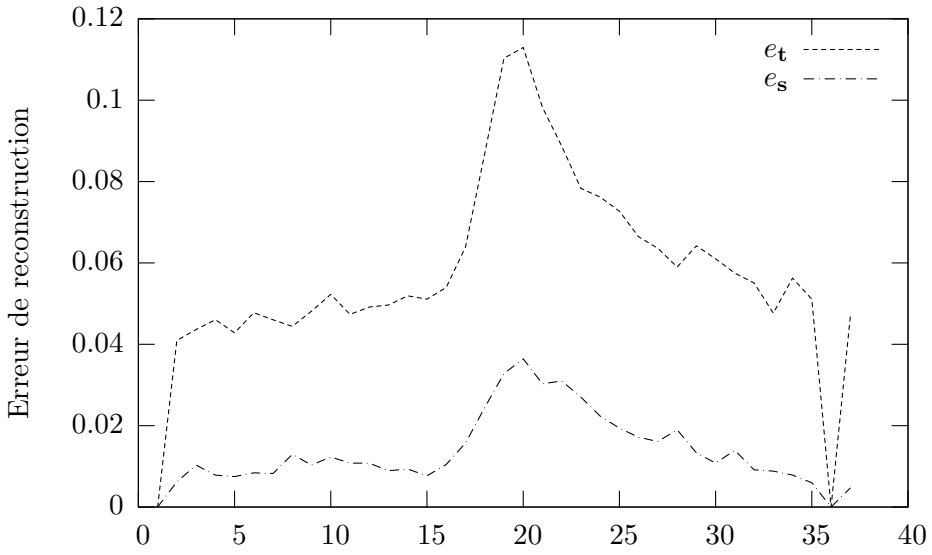


FIG. 6.4 – Erreur de reconstruction de la forme et de la texture au cours de la séquence.

On construit de la même manière une matrice $\mathbf{B_t}$ de textures (il est aussi possible de construire une seule matrice où chaque colonne contient une forme concaténée à une texture). Pour une forme $\mathbf{s}$, le vecteur donnant l'intensité de chacune des expressions est calculé par :

$$\mathbf{b_s} = \mathbf{B_s^+}(\mathbf{s} - \mathbf{s}_n)$$

où $\mathbf{B_s^+}$ désigne la pseudo-inverse de $\mathbf{B}$ ($\mathbf{B}$ doit être de rang N s'il y a N déformations unitaires différentes). Le calcul du vecteur d'intensités à partir des textures est de la même nature.

Certaines déformations faciales sont décrites majoritairement par une variation de forme (mouvement de la bouche ou des sourcils par exemple), d'autres au contraire ne peuvent être décrites que par des variations de texture (saillance de la langue, gonflement des joues, plissement des yeux, etc.). Ainsi, l'intensité des déformations unitaires est déterminée par le maximum entre l'intensité calculée par la matrice de formes $\mathbf{b_s}$ et l'intensité $\mathbf{b_t}$:

$$\mathbf{b} = \max\{\mathbf{b_s}, \mathbf{b_t}\}$$

Nous avons utilisé cette modélisation sur une séquence de 52 images présentant l'*action unit* 36B (joues gonflées par la langue dans sa partie inférieure) jusqu'à son maximum puis, lorsque l'expression est à son pic, un relèvement des sourcils (AU 2), illustrant bien la cooccurrence de deux déformations faciales.

Les matrices $\mathbf{B_s}$ et $\mathbf{B_t}$ ont été construites avec les deux expressions à leur maximum d'intensité.

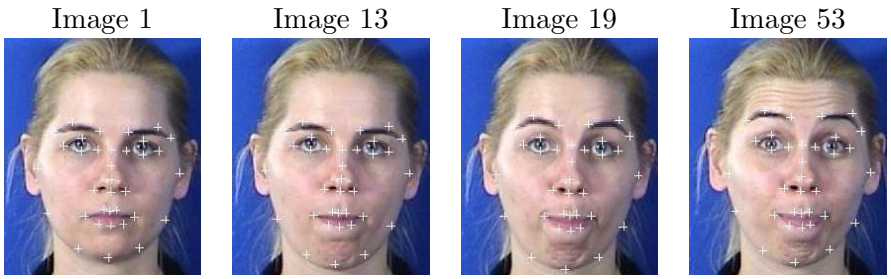


FIG. 6.5 – Images extraites de la séquence « cooccurrence ».

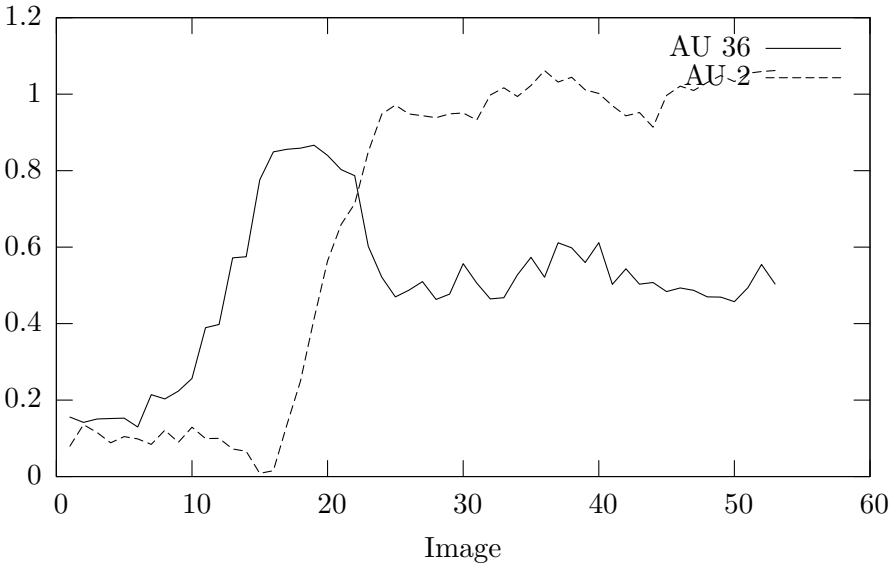


FIG. 6.6 – Évolution de l'intensité des deux expressions (AU36B et AU2) au cours de la séquence vidéo.

La figure 6.6 présente l'évolution de l'intensité au cours de la séquence. Il est à noter que l'évolution des activations détectées suit les activations réelles : l'AU36B est activée, puis l'AU2. Cependant, l'AU36B est détectée comme perdant de son intensité par la suite, alors qu'il n'en est rien en réalité.

6.1.2 Coarticulation d'expressions

L'algorithme d'extraction de l'intensité expressive a de même été appliqué à une séquence contenant deux déformations faciales en coarticulation.

Il s'agit du relèvement des sourcils extérieurs (AU2) et de l'abaissement des sourcils intérieurs (AU4). La figure 2.3 (p. 26) donne un exemple d'une telle coarticulation (avec l'AU1 en plus) que l'on retrouve typiquement dans l'expression de l'émotion de peur.

L'algorithme a été appliqué sur une séquence vidéo représentant l'activation des AU2 et AU4 soit de manière isolée soit en coarticulation. La première sous-séquence (représentée en figure 6.7) correspond à l'activation isolée de l'AU2, puis de l'AU4. Les sous-séquences deux et trois contiennent les AUs 2 et 4 en coarticulation : la deuxième sous-séquence en commençant par l'AU2 et la troisième sous-séquence en commençant par l'AU4.

Précédemment, un AAM a été construit sur quatre images de la séquence vidéo (correspondant à l'expression neutre, au pic d'activation de l'AU2, de l'AU4 et à une coarticulation entre les deux) avec un modèle de forme à 35 points d'intérêt. Les points d'intérêt ont alors été localisés automatiquement sur chacune des images par l'algorithme d'adaptation d'AAM.

Deux vecteurs expressifs ont été construits, pour la mesure de l'activation de l'AU2 et de l'AU4 en soustrayant l'expression neutre à l'activation maximale de l'AU2 et à l'activation maximale de l'AU4. L'image correspondant à la coarticulation de l'AU2 et l'AU4 n'a donc pas été utilisée pour l'extraction de l'intensité d'activation.

Les résultats sont représentés sur les figures 6.7, 6.8 et 6.9. Pour chaque image, l'intensité d'activation des deux déformations faciales est représentée, ainsi que l'erreur de reconstruction de forme $e_{\mathbf{s}}$ et de texture $e_{\mathbf{t}}$.

Les intensités d'activation détectées des deux déformations faciales suivent la même évolution que celles observées sur la séquence vidéo. L'erreur de reconstruction est au maximum d'environ 15% pour la texture et d'environ 6% pour la forme. La forme et la texture ayant la plus forte erreur de reconstruction sont représentées en figure 6.10. Dans la deuxième et troisième sous-séquence la coarticulation est traduite par une des deux activations qui voit son intensité augmenter pendant que l'intensité de l'autre déformation diminue. Il est cependant à noter que les valeurs détectées ne correspondent pas toujours à la réalité. Par exemple, sur l'image 572, l'AU4 est considérée comme ayant une intensité d'activation proche de celle détectée sur l'expression neutre (sur l'image 454 par exemple), alors que ce n'est pas le cas.

6.1.3 Commentaires

Concernant le manque de précision dans la détection de l'intensité d'activation de l'AU36B sur la séquence de cooccurrence, plusieurs sources de bruit peuvent expliquer ce comportement. Premièrement, il est vraisemblable que la précision du placement du modèle ne soit pas optimale. Deuxièmement, les images retenues comme représentant les déformations à leur pic d'intensité peuvent aussi contenir du bruit, dû à de légères différences d'éclairage avec l'image de l'expression neutre ou encore un manque de précision dans le placement du modèle de forme.

Pour éviter une influence trop importante de la deuxième source de bruit, nous proposons de ne retenir dans les images expressives que les parties réellement déformées. Par exemple l'AU36B entraîne une déformation de la partie

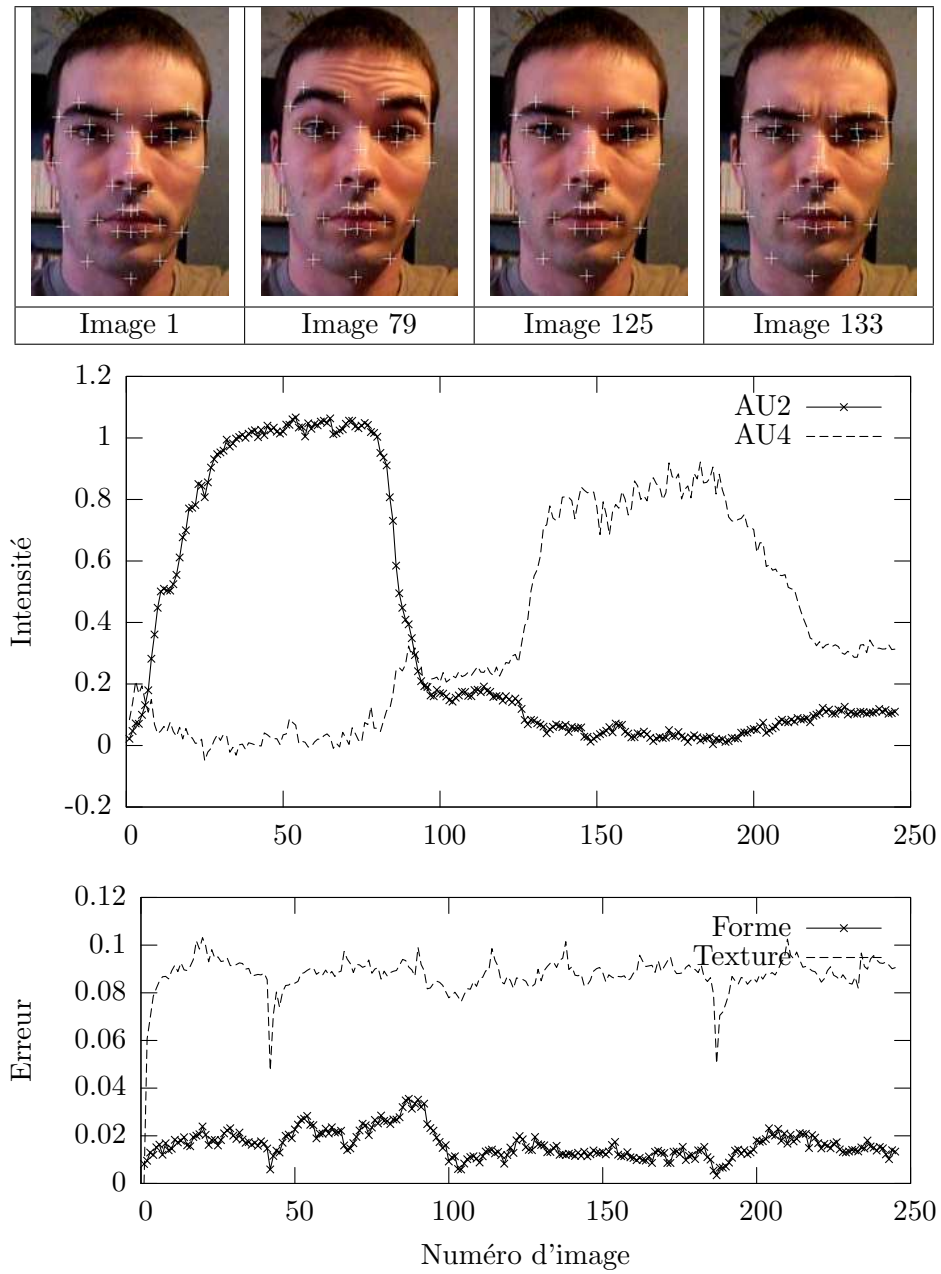


FIG. 6.7 – Extraits de la première sous-séquence de la vidéo « coarticulation », qui correspond à une expression neutre, puis l'AU2, puis un relâchement, puis l'AU4, puis un relâchement. Les courbes du milieu représentent l'évolution de l'intensité d'activation des déformations faciales AU2 et AU4. Les courbes du bas représentent l'erreur de reconstruction de la forme et de la texture.

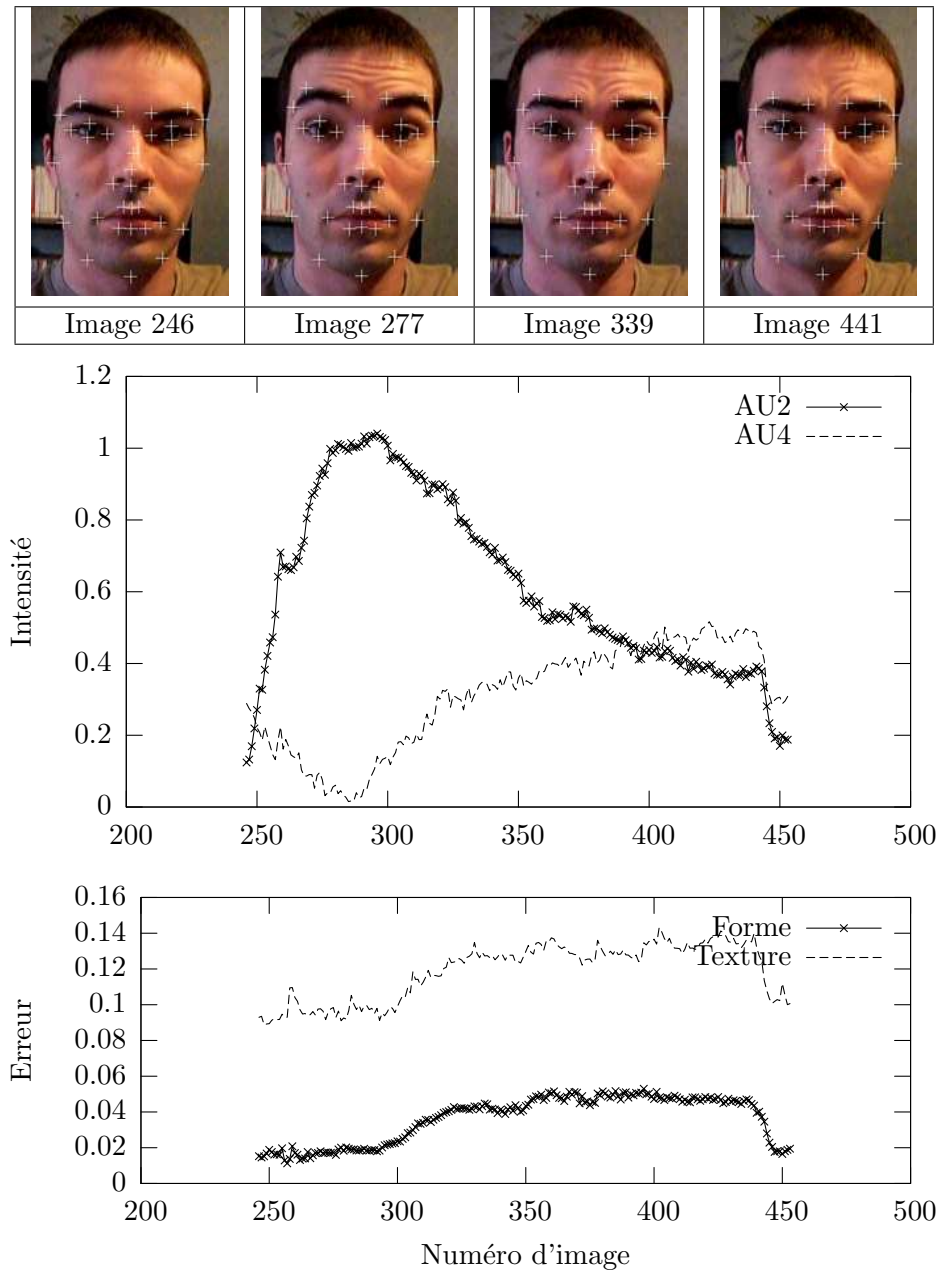


FIG. 6.8 – Extraits de la deuxième sous-séquence de la vidéo « coarticulation », qui correspond à une expression neutre, puis l'AU2, puis l'AU4 en coarticulation, puis un relâchement. Les courbes du milieu représentent l'évolution de l'intensité d'activation des déformations faciales AU2 et AU4. Les courbes du bas représentent l'erreur de reconstruction de la forme et de la texture.

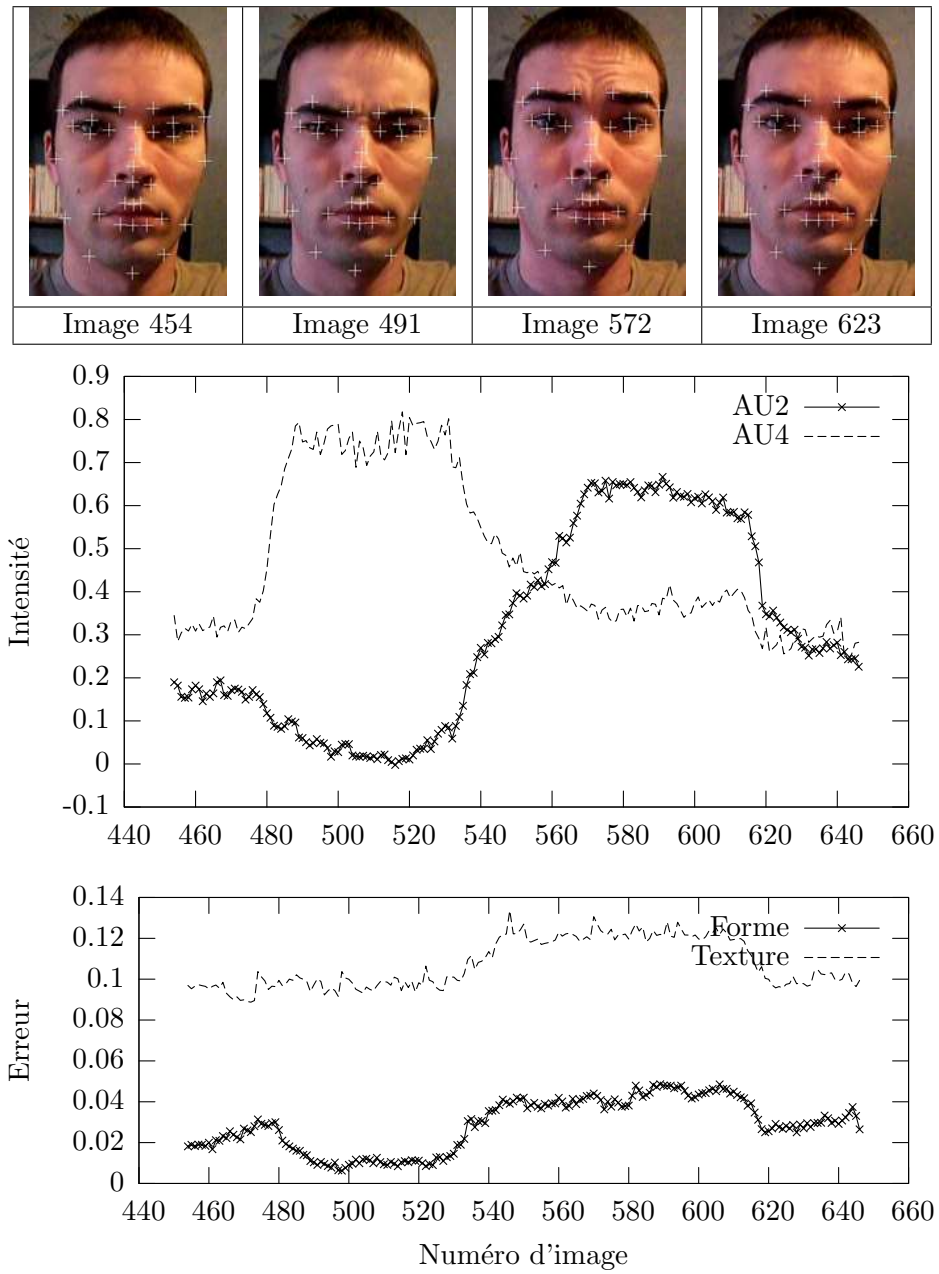


FIG. 6.9 – Extraits de la troisième sous-séquence de la vidéo « coarticulation », qui correspond à une expression neutre, puis l'AU4, puis l'AU2 en coarticulation, puis un relâchement. Les courbes du milieu représentent l'évolution de l'intensité d'activation des déformations faciales AU2 et AU4. Les courbes du bas représentent l'erreur de reconstruction de la forme et de la texture.

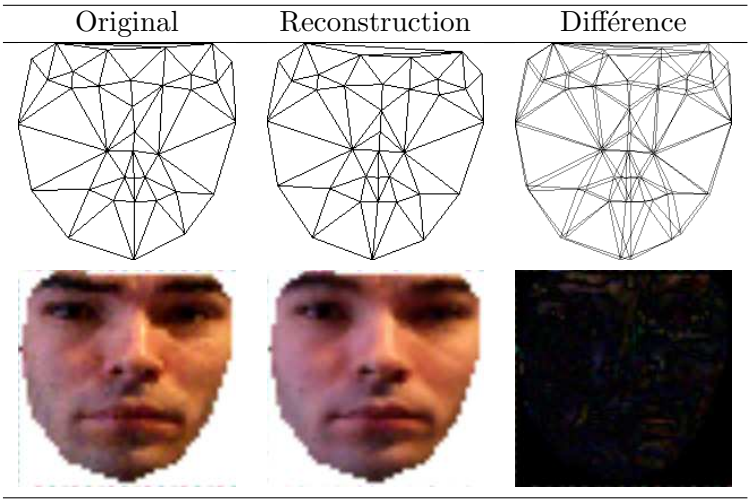


FIG. 6.10 – Visualisation de l'erreur maximale de reconstruction (différence entre l'original et la reconstruction) pour la forme (en haut) et la texture (en bas).

inférieure du menton et de la bouche, mais pas du reste du visage. De même, l'AU2 n'entraîne une déformation que des sourcils et du contour des yeux (ouverture).

On associe à chaque expression, une liste de points de définition $\mathbf{P}$, qui est une sous-partie de l'ensemble des points du modèle de forme. Si $\mathbf{s}_n$ et $\mathbf{t}_n$ représentent la forme et la texture de l'expression neutre et $\mathbf{s}_m$ et $\mathbf{t}_m$ la forme et la texture de l'expression à son pic, alors plutôt que de stocker directement $\mathbf{s}_m$ et $\mathbf{t}_m$ comme définition de l'expression à son pic, on stocke $\hat{\mathbf{s}}_m$ et $\hat{\mathbf{t}}_m$. La forme $\hat{\mathbf{s}}_m$ est définie par les coordonnées de $\mathbf{s}_m$ seulement pour les points de $\mathbf{P}$ et par les points de la forme neutre $\mathbf{s}_n$ pour tous les autres (les points du modèle de forme qui n'appartiennent pas à $\mathbf{P}$). De même, la texture $\hat{\mathbf{t}}_m$ est définie par les pixels de $\mathbf{t}_m$ qui appartiennent aux triangles engendrés par les points de $\mathbf{P}$ et par les pixels de la texture neutre $\mathbf{t}_n$ pour tous les autres triangles.

Nous avons procédé à l'extraction de l'intensité des deux déformations faciales sur la même séquence vidéo que précédemment en utilisant seulement des sous-parties du modèle pour la définition des déformations maximales. Le résultat est représenté sur la figure 6.11.

Il apparaît alors que les résultats sont meilleurs que ceux représentés sur la figure 6.6. En effet, s'agissant de l'AU36, le degré d'activation reste relativement stable après le pic d'activation, ce qui correspond mieux à la réalité.

L'*action unit* 36 est une déformation faciale qui est représentée (en utilisant un modèle 2D) majoritairement par une variation de texture à partir de l'expression neutre. De plus, c'est un cas limite pour la modélisation linéaire, puisque si la langue gonfle la joue à un endroit légèrement différent de celui

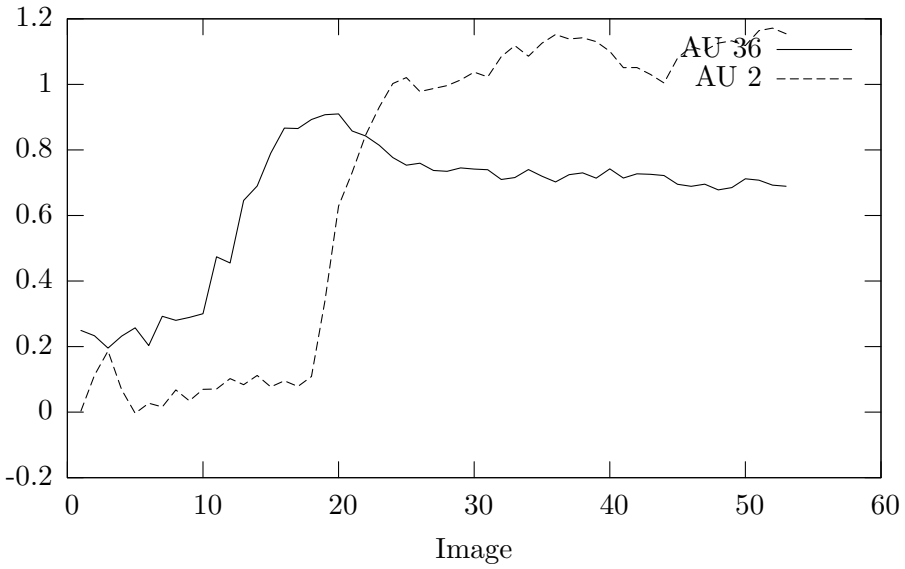


FIG. 6.11 – Intensité détectée des deux déformations (AU36B et AU2) au cours de la séquence vidéo en utilisant un découpage local.

appris, la texture ne pourra pas être représentée efficacement. En effet, une interpolation linéaire de texture peut expliquer des variations d'intensité d'une texture, mais pas des variations de position. Le modèle de forme permet lui à l'inverse d'expliquer des positions différentes d'une même texture. Ainsi, pour prendre en compte le mieux possible une telle déformation faciale, la logique voudrait que le modèle de forme soit augmenté de quelques points, définis par exemple sur les contrastes lors de l'activation de l'AU36. Cependant, la position des points serait très mal définie lorsque l'AU36 n'est pas activée.

Le bruit observé qui subsiste dans la mesure du degré d'activation reflète directement le bruit présent dans les deux images de définition de chaque déformation unitaire. En effet, le modèle de chaque déformation unitaire n'est défini que par une image de l'expression neutre et une image de la déformation à son pic. Les résultats pourraient être améliorés en diminuant le bruit de ces deux images (en utilisant plusieurs images pour le neutre et le pic par exemple).

6.1.4 Évaluation

Une évaluation quantitative du modèle linéaire est difficile à mettre en place. En effet, ceci nécessiterait d'avoir à disposition pour comparaison une vérité terrain. Cette vérité terrain serait constituée d'une séquence d'images expressives auxquelles seraient associée une mesure d'intensité d'activation pour chacune des déformations faciales unitaires. L'intensité d'activation pourrait correspondre, par exemple à un score du système FACS (qui distingue cinq degrés d'activation mesuré par des experts). De plus, il serait nécessaire d'avoir

à disposition l'image de chacune des déformations faciales unitaire à son maximum d'activation, de manière isolée.

Or, il n'existe pas, à notre connaissance, de base de données de séquences vidéo dont les détails d'annotation sont suffisants pour constituer une vérité terrain. La base de vidéos *Cohn-Kanade AU-Coded Facial Expression Database* [Kanade 00] qui est fréquemment utilisée dans le domaine de l'analyse des expressions faciales contient effectivement une annotation FACS réalisée par des experts. Cependant, cette annotation ne comprend que rarement un degré d'activation (et se limite donc à l'indication d'occurrence de chaque AU). De plus quand il est présent, le degré d'activation n'est pas disponible pour chaque image de la séquence vidéo, mais sur le pic de l'expression et les images des déformations unitaires ne sont pas forcément disponibles.

Avec une base de données suffisante, deux tests seraient à mener :

- un test sur le pic d'expressions combinées, pour tester la détection de l'activation des différentes déformations unitaires rentrant en jeu dans la composition de l'expression, sans tenir compte du degré d'activation.
- un test sur séquences expressives annotées, pour tester la détection du degré d'activation des différents déformations unitaires.

FIG. 6.12 – Exemple d'anonymisation par pixelisation d'une image de visage qui dégrade à la fois l'identité et l'expression.

6.2 Anonymisation[3]

La langue des signes ne dispose pas de forme écrite propre. C'est une langue de tradition orale[4]. Ainsi, le support d'échange privilégié est l'enregistrement vidéo. La démocratisation des moyens numériques de communication a vu apparaître un important échange de fichiers vidéos dans la communauté sourde. Dans ce contexte, certains besoins commencent à émerger : en particulier le besoin de pouvoir témoigner de manière anonyme (de la même manière que les échanges textuels des forums de discussion peuvent se faire via l'utilisation de pseudonymes), se rapprochant ainsi d'une des propriétés de la forme écrite.

Les techniques de traitement d'images classiques utilisées pour masquer l'identité d'une personne présente dans une vidéo ne sont pas directement applicables au contexte d'une communication vidéo signée. En effet, ces techniques simples (qui consistent à diminuer grandement la résolution du visage d'un locuteur, en incrustant un flou ou un « mosaïquage », (voir par exemple la figure 6.12) dégradent l'ensemble de l'image du visage : son identité mais aussi ses expressions. Les expressions ayant un rôle prépondérant en LSF, le sens s'en voit lui aussi dégradé.

Afin de remédier à ce problème, nous proposons des techniques qui modifient la partie identitaire d'un visage sans en modifier la partie expressive.

[3]On trouve dans la littérature les deux racines *anonym-* et *anonymis-* pour l'action de rendre quelque chose anonyme. Nous avons choisi les termes *anonymiser* et *anonymisation*, car leur usage est le plus fréquent, bien que l'on trouve des termes avec la racine *anonym-* (à l'Éducation Nationale notamment où l'on *anonyme* des copies).

[4]L'étude de formes graphiques de la langue des signes est un domaine très dynamique, en particulier en France. Les deux partenaires de cette thèse, l'IRIT et WebSourd sont engagés dans un projet national sur ce thème. L'étude des expressions du visage intervient en amont, pour contribuer à la définition de ce formalisme. La description automatique interviendra en aval dans les systèmes d'aide à l'écriture ou à la transcription de la forme « orale » vers la forme écrite.

Le but est ici de tromper le système d'identification humain mais pas son système d'authentification, celui-ci étant extrêmement robuste. Il s'agit de permettre le témoignage de personnes qui ne voudraient pas être reconnues *a posteriori*. Dans le cas du témoignage d'une personne déjà connue, la modification de l'aspect visuel du visage n'empêchera que très difficilement sa reconnaissance. En effet, dans la tâche de vérification d'une identité, le système visuel humain a recours à de nombreux indices autres que le visage : chevelure, vêtements, attitude générale, etc.

Il s'agit donc moins d'empêcher de reconnaître un visage connu que d'empêcher de construire un modèle de visage qui permettrait ensuite de reconnaître la personne.

Nous présentons trois méthodes permettant la modification de l'aspect identitaire d'un visage avec conservation de l'aspect expressif. Ces méthodes sont ensuite évaluées puis appliquées sur une séquence vidéo expressive et comparées qualitativement en terme de qualité visuelle de rendu. Une première étude a été menée sur deux méthodes simples dans [Mercier 05].

Les techniques utilisées fonctionnent en deux temps : dans un premier temps, l'information expressive est extraite de l'image à anonymiser et ce, indépendamment de l'information d'identité ; dans un deuxième temps, cette information expressive est utilisée pour la génération d'un nouveau visage en changeant l'information d'identité. Il s'agit donc de séparer l'aspect expressif de l'aspect identitaire d'une image de visage.

6.2.1 Méthode par translation

Une première modélisation simple consiste à considérer la forme (ou la texture) d'un visage comme étant une forme (ou texture) neutre en expression (mais spécifique à l'identité) à laquelle est ajoutée une somme pondérée de déformations (ou variations de texture) spécifiques à l'expression (mais pas à l'identité).

Ainsi, pour une forme d'identité i et d'expression e, on a (cette modélisation est inspirée de [Costen 02]) :

$$\mathbf{p}^{i,e} = \mathbf{p}_n^i + \sum_{j=1}^{Ne} \mathbf{v}^j b_j$$

ou, sous forme matricielle :

$$\mathbf{p}^{i,e} = \mathbf{p}_n^i + \mathbf{V}\mathbf{b}^e$$

On suppose alors cette dernière équation vérifiée pour toutes les images d'un ensemble de $Ni \times Ne$ visages, où Ni est le nombre d'identités différentes et Ne le nombre d'expressions différentes.

Le but est alors de calculer la matrice $\mathbf{V}$ sur la base de visages. Il existe une infinité de solutions pour le choix de $\mathbf{V}$. Nous ajoutons alors une contrainte

d'orthonormalité sur les colonnes de $\mathbf{V}$ et la solution est obtenue par une analyse en composantes principales sur la matrice de covariance $\mathbf{C}$ suivante :

$$\mathbf{C} = \frac{1}{NiNe} \sum_{i=1}^{Ni} \sum_{j=1}^{Ne} (\mathbf{p}^{i,j} - \mathbf{p}_n^i)(\mathbf{p}^{i,j} - \mathbf{p}_n^i)^T$$

Pour un visage d'identité i, on peut alors extraire les paramètres expressifs :

$$\mathbf{b} = \mathbf{V}^T(\mathbf{p} - \mathbf{p}_n^i)$$

On peut alors effectuer un changement de l'identité i vers l'identité j avec :

$$\hat{\mathbf{p}} = \mathbf{V}\mathbf{b} + \mathbf{p}_n^j$$

6.2.2 Méthode par factorisation

Dans [Abboud 04], est présentée une modélisation permettant de décorréler identité et expression. Il s'agit de considérer qu'un visage (représenté ici par un vecteur de déformations dans l'espace des formes et textures, obtenues précédemment par une analyse en composantes principales) peut être représenté par l'interaction entre un vecteur de déformations spécifique à l'expression et une matrice spécifique à l'identité :

$$\mathbf{p}_{i,e} = \mathbf{A}^i \mathbf{b}^e$$

(on considère $\mathbf{p}$ comme désignant les paramètres de forme $\mathbf{p}^s$ ou les paramètres de texture $\mathbf{p}^t$.)

Le but est d'apprendre une matrice $\mathbf{A}^i$ pour chaque identité i différente. On suppose une base de visages contenant n identités différentes affichant chacune m expressions. Chaque visage d'identité i et d'expression e est codé par un vecteur de forme $\mathbf{p}_{i,e}^s$ et un vecteur de texture $\mathbf{p}_{i,e}^t$. On construit une matrice $\mathbf{C}$ par concaténation des paramètres de chacun des visages :

$$\mathbf{C} = \begin{bmatrix} \mathbf{p}_{1,1} & \cdots & \mathbf{p}_{1,m} \\ & \cdots & \\ \mathbf{p}_{n,1} & \cdots & \mathbf{p}_{n,m} \end{bmatrix}$$

On suppose que $\mathbf{C}$ est le résultat de la multiplication d'une matrice $\mathbf{\Gamma}$ (empilement des $\mathbf{A}^i$) par une matrice $\mathbf{B}$ (empilement des $\mathbf{b}^e$) :

$$\mathbf{C} = \mathbf{\Gamma}\mathbf{B}$$

Les matrices $\mathbf{\Gamma}$ et $\mathbf{B}$ peuvent être obtenues par une décomposition en valeurs singulières (SVD). Si $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, alors :

$$\mathbf{\Gamma} = \mathbf{U}\sqrt{\mathbf{\Sigma}}$$

$$\mathbf{B} = \sqrt{\mathbf{\Sigma}}\mathbf{V}^T$$

Afin de garder un maximum d'information, on considère que les espaces de formes et de textures ont été construits en retenant suffisamment de vecteurs pour expliquer 100% de la variance (correspondant à $N-1$ vecteurs si les espaces ont été construits à partir de N exemples indépendants). De même, lors de l'utilisation de la SVD, on retient le maximum de colonnes possibles pour $\mathbf{\Gamma}$ et le maximum de lignes possible pour $\mathbf{B}$ (c'est à dire autant que le nombre d'expressions dans la base d'apprentissage).

On extrait alors les matrices $\mathbf{A}^i$ de $\mathbf{\Gamma}$, chacune de taille $(N-1) \times m$. Pour un visage de l'identité i, il est alors possible d'extraire les paramètres expressifs par :

$$\mathbf{b} = (\mathbf{A}^i)^+ \mathbf{p}$$

Un changement d'identité de i à j peut alors être effectué par :

$$\hat{\mathbf{p}} = \mathbf{A}^j \mathbf{b}$$

Note : Les calculs sont ici effectués sur les paramètres de forme et de texture dans les sous-espaces vectoriels obtenus par analyse en composantes principales sur l'ensemble des images de la base d'apprentissage. Théoriquement, rien n'empêche d'utiliser directement les formes et les textures (normalisées). Cependant, les vecteurs résultants étant de taille très importante, le calcul de la SVD, bien qu'effectué hors-ligne, devient très coûteux.

6.2.3 Méthode par projection

La troisième méthode consiste à appliquer la modélisation présentée en début de chapitre qui permet d'extraire l'intensité d'activation de chacune des déformations faciales.

L'intensité d'activation est obtenue par projection sur la base des déformations : la base est centrée sur le visage neutre en expression et chacun des vecteurs de la base correspond à la différence entre le visage à son pic d'expression et le visage neutre. Pour l'identité i, la base expressive des textures est :

$$\mathbf{B}_\mathbf{t}^i = \left[(\mathbf{t}_{m,1}^i - \mathbf{t}_n^i), \quad \ldots, \quad (\mathbf{t}_{m,N}^i - \mathbf{t}_n^i)\right]$$

Il est possible de calculer une telle base pour une autre identité j.

$$\mathbf{B}_\mathbf{t}^j = \left[(\mathbf{t}_{m,1}^j - \mathbf{t}_n^j), \quad \ldots, \quad (\mathbf{t}_{m,N}^j - \mathbf{t}_n^j)\right]$$

Sur une image expressive de l'identité i, on extrait les paramètres d'expression :

$$\mathbf{b}_\mathbf{t} = (\mathbf{B}_\mathbf{t}^i)^+ (\mathbf{t} - \mathbf{t}_n^i)$$

Le changement d'identité est effectué par :

$$\hat{\mathbf{t}} = \mathbf{B}_\mathbf{t}^j \mathbf{b}_\mathbf{t} + \mathbf{t}_n^j$$

6.2.4 Mise en œuvre

Nous avons mis en œuvre ces méthodes pour modifier l'identité d'un visage d'une séquence expressive (extraite de la base MMI [Pantic 05]).

La base d'apprentissage a été construite sur la base de vidéos expressives MMI, en sélectionnant 18 déformations faciales :

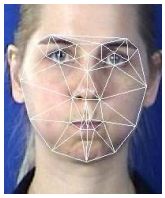
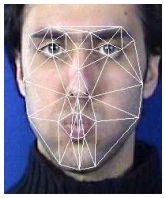
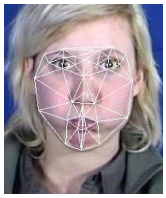
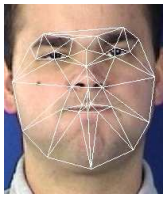
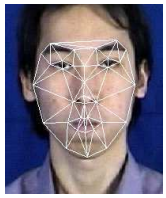
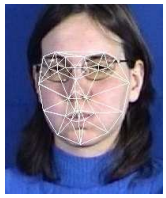
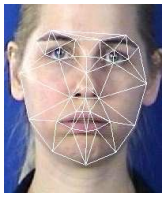
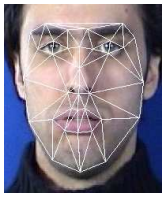
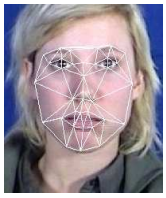
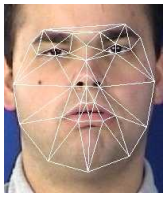
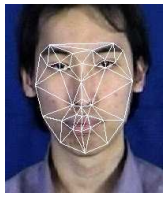
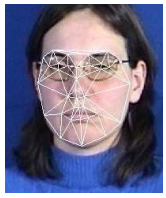
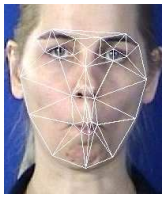
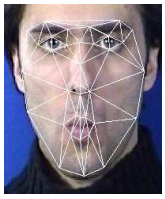
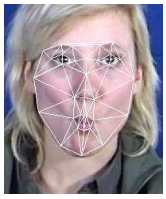
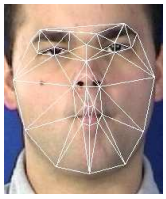
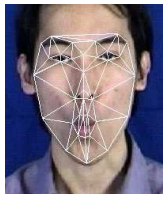
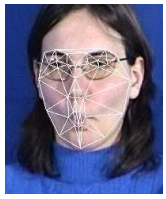
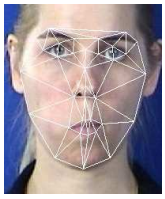
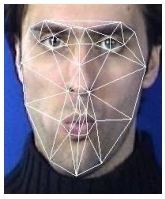
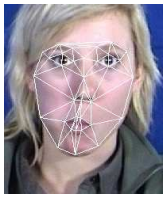
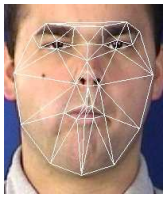
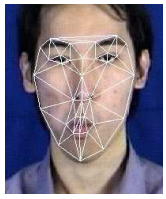
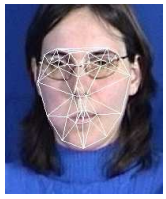
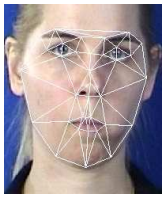
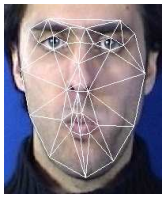
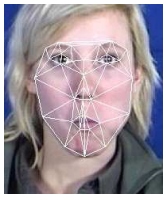
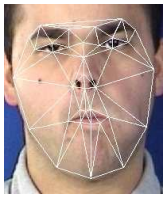
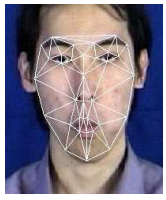
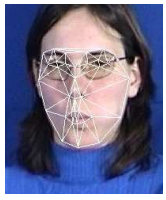
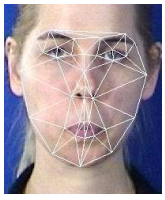
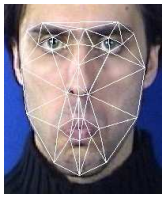
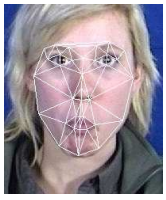
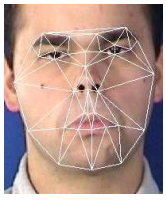
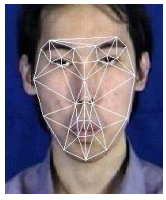
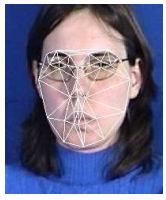
Description	Action Unit
Joues gonflées	AU 34
Souffle	AU 33
Joues pincées	AU 35
Langue saillante L	AU 36L
Langue saillante R	AU 36R
Langue saillante T	AU 36T
Langue saillante B	AU 36B
Sourire	AU 12 (+6)
Sourire avec dents visibles	AU 12 (+6)
Yeux plissés	AU 7 (+6)
Yeux fermés	AU 43
Yeux grand ouverts	AU 5
Sourcils relevés	AU 2
Moue	AU 15
Bouche ouverte	AU 26 (/27)
Langue visible	AU 19
Lèvres avancées	AU 18
Lèvres pincées	AU 28

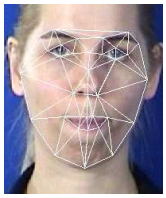
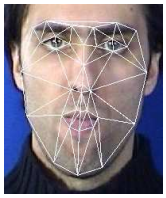
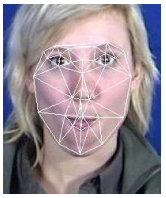
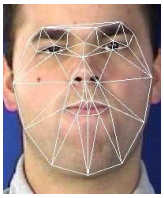
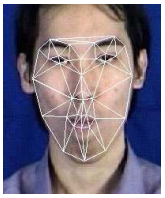
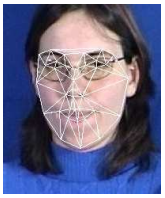
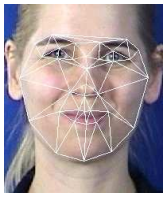
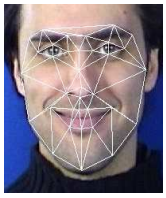
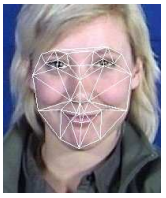
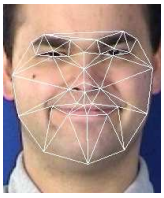
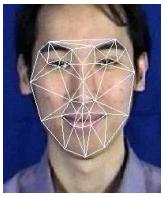
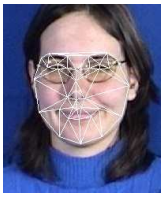
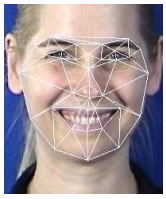
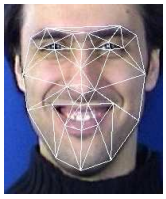
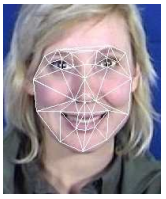
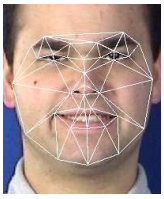
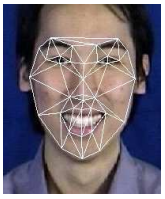
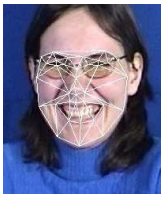
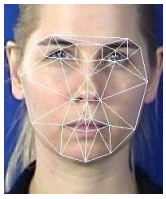
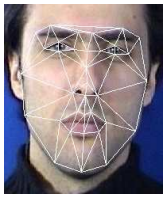
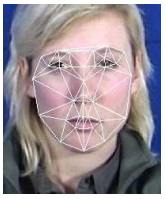
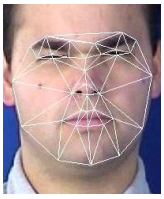
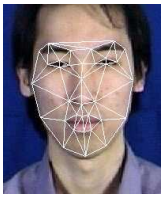
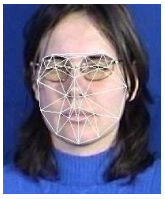
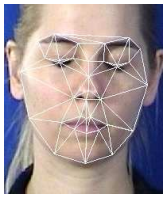
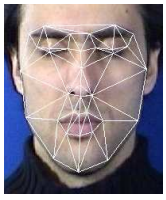
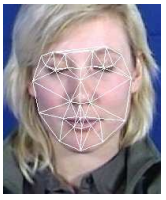
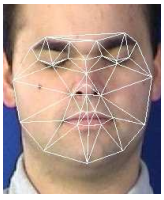
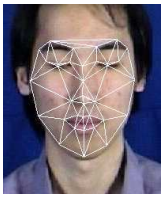
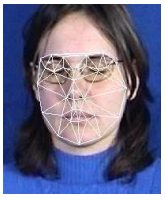
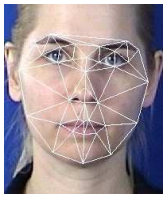
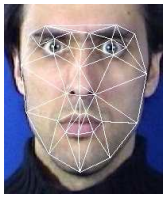
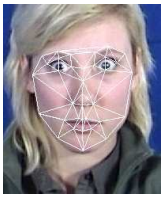
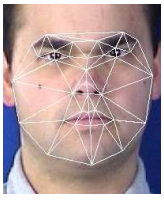
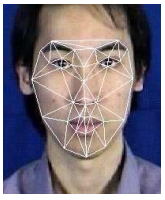
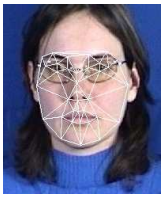
Nous avons retenu les vidéos de 6 identités différentes (comprenant 3 femmes et trois hommes d'origines différentes) produisant chacune de ces déformations en partant de l'expression neutre. Pour chaque déformation, l'image correspondant au maximum d'intensité a été extraite et 35 points d'intérêt ont été positionnés manuellement. L'ensemble de la base est représenté en figure 6.13.

La vidéo à analyser contient la première identité. Le but est d'extraire le mouvement des points d'intérêt sur cette vidéo puis d'appliquer une méthode d'anonymisation.

Dans un premier temps, nous avons donc construit un AAM spécifique à l'identité 1, en ne retenant pour ce faire que les expressions de cette identité pour la construction du modèle. Suffisamment de vecteurs de forme et de texture ont été retenus pour expliquer 95% de la variance. Nous avons utilisé l'algorithme simultané pour suivre les déformations faciales au cours de la vidéo.

Pour la méthode à base de factorisation, une analyse en composantes principales a été calculée sur l'ensemble des 6×18 expressions (et l'expression neutre) en retenant 100% de la variance. Cette analyse servant à la procédure

E	Identité 1	Identité 2	Identité 3	Identité 4	Identité 5	Identité 6
1						
2						
3						
4						
5						
6						

E	Identité 1	Identité 2	Identité 3	Identité 4	Identité 5	Identité 6
7						
8						
9						
10						
11						
12						

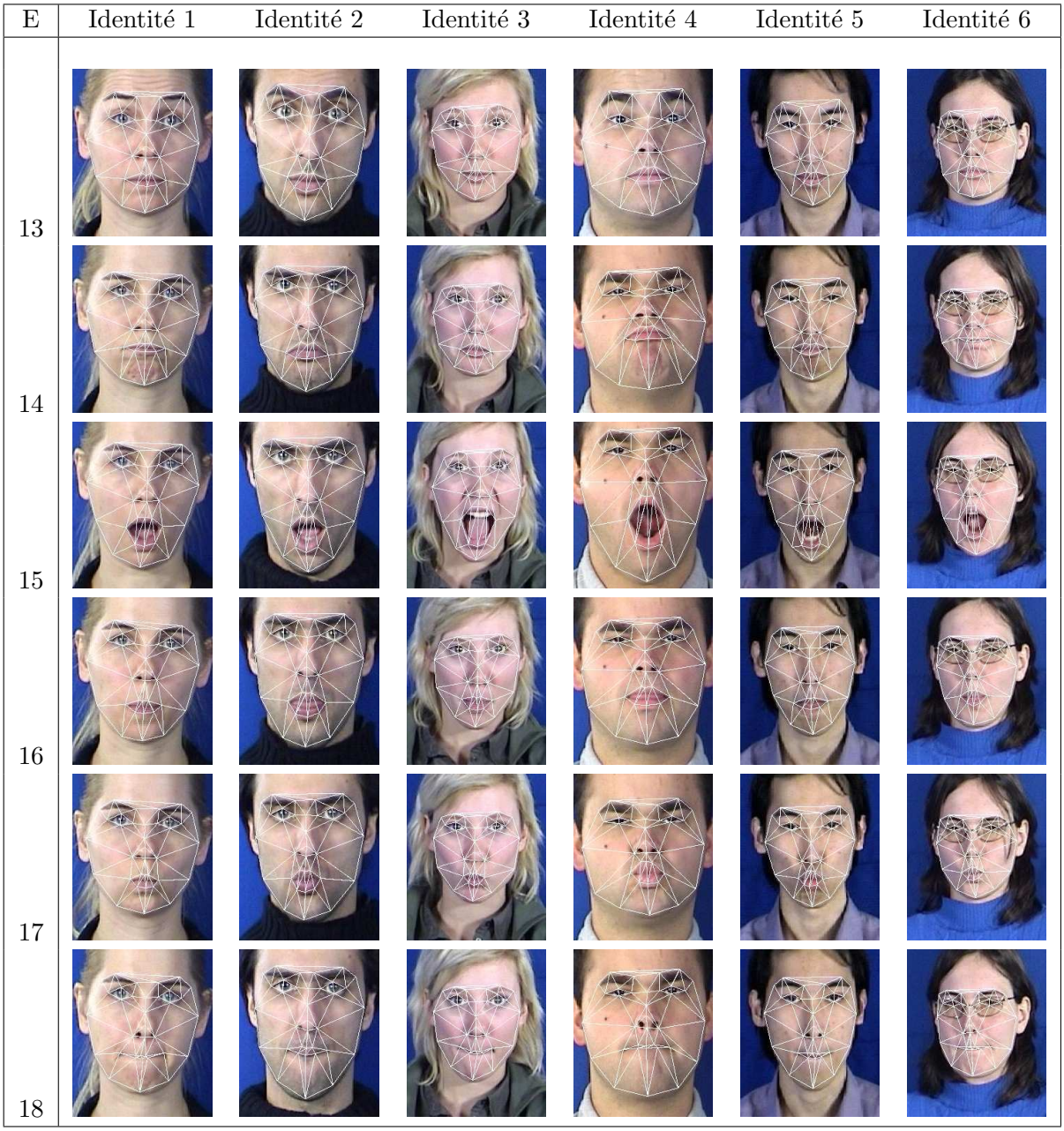


FIG. 6.13 – Extrait retenu de la base MMI

d'anonymisation et non à l'algorithme d'adaptation d'AAM, il est possible de retenir un grand nombre de vecteurs. De plus, nous avons gardé les trois canaux de couleur pour construire les textures : un vecteur de texture est constitué par la concaténation des trois vecteurs de texture de chacun des canaux.

Les matrices $\mathbf{A}^i$ de la méthode par factorisation ont été obtenues soit par une SVD sur une matrice contenant les 6 identités (et 19 expressions) de MMI, soit seulement les deux identités (l'identité originale et l'identité servant à anonymiser).

Le traitement pour les deux méthodes d'anonymisation a consisté à travailler dans un premier temps uniquement sur la texture. La forme n'a pas été modifiée, étant donné que l'information la plus spécifique à l'identité est la texture.

Dans les deux cas, la texture a été échantillonnée sur une forme moyenne $\mathbf{s}_0$ de résolution 48×48.

La vidéo de test est une vidéo où l'identité 1 affiche un certain nombre d'expressions. Certaines ne font pas partie de la base d'apprentissage de l'AAM. Ainsi, l'adaptation du modèle déformable n'est pas optimale sur certaines images, ce qui permet de mesurer le comportement des algorithmes d'anonymisation dans un tel cas.

6.2.5 Évaluation de l'anonymisation

La procédure d'anonymisation est menée ici par le changement d'une identité à une autre sans modification de l'expression. Soit un visage d'identité i et d'expression e anonymisé vers un visage d'identité j. Si on y applique un système de reconnaissance d'expression (humain ou automatique), il doit retourner l'expression e. Et si un système de reconnaissance d'identité y est appliqué, il doit retourner l'identité j (et non i).

Test préalable

Nous avons à notre disposition une base de Ni identités affichant chacune Ne mêmes expressions. Un algorithme d'anonymisation modifie une image $I_{i,e}$ d'identité i et d'expression e en une image $\hat{I}_{j,e}$de même expression et d'identité différente j. Or, dans la base de visages, puisque les expressions sont en correspondance, on dispose de l'image réelle $I_{j,e}$ d'identité j et d'expression e. Une première évaluation consiste donc à mesurer la différence entre l'image anonymisée $\hat{I}_{j,e}$ et l'image réelle $I_{j,e}$.

Pour chaque image $I_{i,e}$ d'identité i et d'expression e
    Calculer l'anonymisation $\hat{I}_{j,e}$, avec $\forall j = 1 \dots Ni$.
    Calculer $D_{i,e,j} = ||I_{j,e} - \hat{I}_{j,e}||$
Fin pour
Calculer $E_{i,e} = \max_j \{D_{i,e,j}\}/N_0$,
    où N_0 est la résolution d'échantillonnage de $\mathbf{s}_0$

Expression \ Identité	**1**	**2**	**3**	**4**	**5**	**6**
1	1.3	1.3	1.5	1.3	1.3	1.5
2	1.0	1.1	1.1	1.1	1.1	1.1
3	1.1	1.1	1.1	1.1	1.1	1.1
4	1.1	1.2	1.3	1.1	1.2	1.3
5	1.0	1.1	1.1	1.1	1.1	1.1
6	1.3	1.1	1.3	1.2	1.1	1.2
7	1.2	1.1	1.2	1.2	1.1	1.2
8	1.3	1.2	1.3	1.2	1.2	1.3
9	1.4	1.5	1.3	1.4	1.5	1.5
10	1.1	1.1	1.3	1.2	1.1	1.3
11	1.2	1.2	1.3	1.1	1.2	1.3
12	1.2	1.3	1.4	1.3	1.2	1.4
13	1.3	1.4	1.4	1.5	1.4	1.5
14	1.0	1.1	1.2	1.1	1.1	1.2
15	1.2	1.2	1.4	1.3	1.2	1.4
16	1.2	1.1	1.3	1.3	1.3	1.3
17	1.2	1.1	1.3	1.3	1.2	1.3
18	1.3	1.1	1.3	1.2	1.3	1.3

TAB. 6.1 – Erreur entre le visage anonymisé $\hat{I}_{j,e}$ et le visage de la base $I_{j,e}$, par pixel de texture, pour chaque identité et expression de la base. L'anonymisation est effectuée avec la méthode par **translation**.

La table 6.1 représente la valeur de $E_{i,e}$ pour chaque identité et expression de la base lorsque la méthode d'anonymisation utilisée est la méthode par translation. Il se trouve que les deux autres méthodes, par projection et par factorisation donnent des distances rigoureusement nulles pour toutes les identités et toutes les expressions. Ceci vient du fait que, par construction, $\hat{I}_{j,e}$ est égale à $I_{j,e}$ pour la méthode par projection et par factorisation.

Ce test permet de vérifier que les méthodes d'anonymisation appliquées à la base d'apprentissage sont bien définies. En effet, même pour la méthode par translation, l'erreur de texture est pratiquement nulle.

Non-reconnaissance de l'identité

Comme nous l'avons vu dans le chapitre sur l'état de l'art des méthodes d'analyse du visage, une approche classique pour la reconnaissance d'identité consiste à considérer les visages comme faisant partie d'un espace vectoriel sur lequel il est possible de définir une distance (généralement euclidienne). L'apprentissage consiste alors à former Ni classes d'identité et à calculer pour une nouvelle image sa distance à chacune des classes (à chacun des centres des

classes par exemple).

Dans la technique présentée par Turk *et al.*[Turk 91], les image sont toutes considérées de la même dimension et centrées. En pratique, il sera montré que l'alignement des images est le point noir de cette technique (voir [Martinez 02] par exemple). Dans notre cas, nous disposons d'un bon alignement puisque nous disposons de la localisation d'un ensemble de points d'intérêt sur chacune des images.

La reconnaissance de visage consiste, dans notre cas, à calculer une distance euclidienne entre la texture du visage à reconnaître et la texture de chaque identité et à retourner la classe dont la texture est la plus proche.

Pour évaluer dans quelles mesures la reconnaissance d'un visage anonymisée est rendue difficile, nous utilisons un algorithme de reconnaissance d'identité sur notre base de visages. Nous disposons de $Ni = 6$ identités qui sont définies chacune pour $Ne = 18$ images expressives. Dans ces conditions, l'algorithme de reconnaissance d'identités consiste en une tâche de classification parmi $Ni = 6$ classes.

Le nombre d'identités étant faible, nous avons augmenté la base des identités avec un ensemble de 37 identités extraites de la base de visages IMM [Fagertun 05], chacune définie par 3 images (neutre en expression avec un éclairage global, neutre en expression avec un éclairage de côté et affichant un sourire avec un éclairage global). L'ensemble de ces images a été préalablement segmenté manuellement[5]. Nous avons donc un total de $Nt = 43$ identités pour tester la reconnaissance.

Nous calculons dans un premier temps le centre de chacune des classes d'identité C_i (pour $i = 1 \dots Nt$) par une moyenne.

Le protocole d'évaluation est le suivant :

Pour chaque image $I_{i,e}$ d'identité i et d'expression e
Calculer l'anonymisation $\hat{I}_{j,e}$, avec $\forall j = 1 \dots Ni$.
Lancer la reconnaissance d'identité et stocker

$$r_{i,e,j} = \arg\min_{k=1\dots Nt} ||\hat{I}_{j,e} - C_k||$$

Fin pour

Calculer $R_{i,j} = \dfrac{1}{Ne} \displaystyle\sum_{e=1}^{Ne} \delta(r_{i,e,j}, j)$
(où $\delta(i, j)$ est le symbole de Kronecker)

Les tables 6.2, 6.3 et 6.4 représentent les valeurs de $R_{i,j}$ pour les trois méthodes d'anonymisation. Les résultats de reconnaissance d'identité sont très bons, indiquant qu'une image représentant un visage d'identité i modifiée vers une identité j est presque toujours reconnu comme étant d'identité j. De plus,

[5]L'ensemble de la base de visages avec ségmentation peut être trouvé sur le site de Mikkel B. Stegmann - `http://www2.imm.dtu.dk/~aam/`. Quelques modifications ont cependant été apportées au modèle de forme qui contenait initialement 58 points, de manière à le rendre compatible avec notre modèle de forme à 35 points.

j \ i	**1**	**2**	**3**	**4**	**5**	**6**
1		94.4	94.4	94.4	94.4	94.4
2	100		100	100	100	100
3	100	100		100	100	100
4	100	100	100		100	100
5	100	100	100	100		100
6	100	100	100	100	100	

TAB. 6.2 – Taux de reconnaissance de chaque identité modifiée par la procédure d'anonymisation par **projection**, quelque soit l'expression. La reconnaissance est positive quand l'image $I_{i,e}$ modifiée en $\bar{I}_{j,e}$ est reconnue comme étant d'identité j. Le taux de reconnaissance globale est de **99.2%**

j \ i	**1**	**2**	**3**	**4**	**5**	**6**
1		94.4	94.4	94.4	94.4	94.4
2	100		100	100	100	100
3	100	100		100	100	100
4	100	100	100		100	100
5	100	100	100	100		100
6	100	100	100	100	100	

TAB. 6.3 – Taux de reconnaissance de chaque identité modifiée par la procédure d'anonymisation par **factorisation**, quelque soit l'expression. La reconnaissance est positive quand l'image $I_{i,e}$ modifiée en $\bar{I}_{j,e}$ est reconnue comme étant d'identité j. Le taux de reconnaissance globale est de **99.2%**

après investigation sur les quelques cas d'erreurs, il s'avère que l'identité j est confondue avec une autre identité différente de l'identité d'origine i.

Nous en concluons donc que vis-à-vis d'une méthode naïve de reconnaissance d'identité, les trois procédures d'anonymisation modifient de manière efficace le visage d'origine de telle sorte qu'il est impossible à reconnaître sur une base de 43 identités.

Reconnaissance de l'expression

Le but d'un algorithme d'anonymisation est d'empêcher l'identification d'identité. Mais dans notre cas, il s'agit aussi de permettre la reconnaissance d'expression.

Nous utilisons la même procédure que pour la reconnaissance d'identité, excepté le fait que la base initiale n'est pas augmentée de nouvelles expressions.

Contrairement à la reconnaissance d'identité qui n'utilisait que les données

j \ i	**1**	**2**	**3**	**4**	**5**	**6**
1		100	94.4	100	100	100
2	94.4		100	100	100	100
3	100	100		100	100	100
4	100	100	100		100	100
5	100	100	100	100		94.4
6	100	100	100	100	100	

TAB. 6.4 – Taux de reconnaissance de chaque identité modifiée par la procédure d'anonymisation par **translation**, quelque soit l'expression. La reconnaissance est positive quand l'image $I_{i,e}$ modifiée en $\bar{I}_{j,e}$ est reconnue comme étant d'identité j. Le taux de reconnaissance globale est de **99.5%**

de texture, la reconnaissance d'expressions est faite sur les données de texture et de forme. En effet, certaines expressions ne différent que par leur variation de forme par rapport à la forme neutre en expression et très peu par une variation de texture comme c'était le cas pour l'identité.

La forme est constituée de coordonnées 2D d'un ensemble de points d'intérêt. Les formes sont alignées entres elles (en particulier les similarités euclidiennes ont été annulées).

Les données de forme entrent en jeu dans les calculs généralement par l'augmentation du vecteur de texture : là où le calcul s'effectuait avec un vecteur de texture, il s'effectue maintenant avec le même vecteur auquel on a concaténé la forme. Les coordonnées des formes sont recalculées dans l'intervalle $[0, 255]$ de manière à être compatible avec les pixels de texture. De plus, lorsqu'il s'agit de calculer la norme d'un vecteur texture-forme, on pèse différemment la partie texture de la partie forme. En particulier, la norme du vecteur $\mathbf{v}$, résultat de la concaténation $[\mathbf{v}^t, \mathbf{v}^s]$ est calculée par :

$$\sqrt{\mathbf{v}^T \mathbf{A} \mathbf{v}}$$

avec $\mathbf{A}$ matrice diagonale, constituée sur sa diagonale de :

$$\mathbf{a}_i = \begin{cases} 1 & \text{si } i \leq Np \\ \frac{Np}{2Nv} & \text{sinon} \end{cases}$$

où Np est le nombre de pixels de la texture et Nv le nombre de points de la forme.

La reconnaissance d'expressions consiste donc, de manière symétrique à la reconnaissance d'identité, à calculer dans un premier temps la définition de chaque expression L_e de la base en moyennant chaque image d'identité i et d'expression e sur toutes les identités. Dans un deuxième temps, on calcule une différence entre l'image du visage dont on veut reconnaître l'expression et

chaque classe L_e (on utilise dans ce cas une norme pondérée comme présenté précédemment).

Pour chaque image $I_{i,e}$ d'identité i et d'expression e
Calculer l'anonymisation $\hat{I}_{j,e}$, avec $\forall j = 1 \dots Ni$.
Lancer la reconnaissance d'expression et stocker

$$r_{i,e,j} = \arg\min_{k=1\dots Ne} ||\hat{I}_{j,e} - L_k||$$

Fin pour

Calculer $T_{i,e} = \frac{1}{Ni} \sum_{j=1}^{Ni} \delta(r_{i,e,j}, e)$
(où $\delta(i, e)$ est le symbole de Kronecker)

Nous testons au préalable l'efficacité de cet algorithme, en présentant à l'algorithme de reconnaissance chaque image de la base d'apprentissage, sans modification. Ainsi une image $I_{i,e}$ d'identité i et d'expression e doit être reconnue comme affichant l'expression e.

Après calculs, le taux de reconnaissance d'expression sur la base d'apprentissage contenant $6 \times 18 = 108$ visages est de **69%**.

Ce résultat est mauvais, comparé aux taux de reconnaissance d'identité. Ceci s'explique par le fait qu'entre deux images de visages (et même en ajoutant la forme), la différence entre deux identités est généralement plus importante que la différence entre deux expressions.

Néanmoins, connaissant ce taux de reconnaissance de référence, il nous est possible de le comparer au taux de reconnaissance obtenu après application de chacun des algorithmes d'anonymisation.

Les tables 6.5, 6.6 et 6.7 résument les résultats de reconnaissance après modification des images par chacun des algorithmes d'anonymisation. On notera que les méthodes d'anonymisation par projection et par factorisation donnent des résultats rigoureusement égaux entre eux et un taux global identique au taux global de référence. Il apparaît alors que seul l'application de la méthode par translation donne de moins bons résultats de reconnaissance d'expressions.

Évaluation sur une séquence vidéo

L'évaluation menée précédemment ne prenait en considération que des images qui font partie de la base d'apprentissage qui a permis de construire chacune des méthodes.

Nous effectuons les tests de reconnaissance d'identité sur chaque image de la vidéo utilisée en 6.2.4. La reconnaissance d'expression n'aurait pas de sens dans ce cas, puisque l'expression ne peut plus être réduite à une classe parmi Ne, mais combinée à chaque instant de plusieurs expressions unitaires. Il serait éventuellement possible d'extraire les paramètres expressifs sur chacune des images en ayant recours à une des méthodes. Mais nous ne disposons pas de vérité terrain pour vérifier que les paramètres expressifs extraits des images

Expression \ Identité	**1**	**2**	**3**	**4**	**5**	**6**
1	0.5	0.5	0.5	0.5	0.5	0.5
2	0.7	0.7	0.7	0.7	0.7	0.7
3	0.3	0.3	0.3	0.3	0.3	0.3
4	0.5	0.5	0.5	0.5	0.5	0.5
5	0.7	0.7	0.7	0.7	0.7	0.7
6	0.7	0.7	0.7	0.7	0.7	0.7
7	0.7	0.7	0.7	0.7	0.7	0.7
8	0.8	0.8	0.8	0.8	0.8	0.8
9	0.8	0.8	0.8	0.8	0.8	0.8
10	0.5	0.5	0.5	0.5	0.5	0.5
11	1.0	1.0	1.0	1.0	1.0	1.0
12	0.7	0.7	0.7	0.7	0.7	0.7
13	0.7	0.7	0.7	0.7	0.7	0.7
14	0.7	0.7	0.7	0.7	0.7	0.7
15	1.0	1.0	1.0	1.0	1.0	1.0
16	0.7	0.7	0.7	0.7	0.7	0.7
17	0.8	0.8	0.8	0.8	0.8	0.8
18	0.8	0.8	0.8	0.8	0.8	0.8

TAB. 6.5 – Taux de reconnaissance d'expression pour chaque image de la base, modifiée par l'algorithme d'anonymisation par **projection**, quelque soit l'identité cible. Le taux global de reconnaissance est de **69%**.

Expression \ Identité	**1**	**2**	**3**	**4**	**5**	**6**
1	0.5	0.5	0.5	0.5	0.5	0.5
2	0.7	0.7	0.7	0.7	0.7	0.7
3	0.3	0.3	0.3	0.3	0.3	0.3
4	0.5	0.5	0.5	0.5	0.5	0.5
5	0.7	0.7	0.7	0.7	0.7	0.7
6	0.7	0.7	0.7	0.7	0.7	0.7
7	0.7	0.7	0.7	0.7	0.7	0.7
8	0.8	0.8	0.8	0.8	0.8	0.8
9	0.8	0.8	0.8	0.8	0.8	0.8
10	0.5	0.5	0.5	0.5	0.5	0.5
11	1.0	1.0	1.0	1.0	1.0	1.0
12	0.7	0.7	0.7	0.7	0.7	0.7
13	0.7	0.7	0.7	0.7	0.7	0.7
14	0.7	0.7	0.7	0.7	0.7	0.7
15	1.0	1.0	1.0	1.0	1.0	1.0
16	0.7	0.7	0.7	0.7	0.7	0.7
17	0.8	0.8	0.8	0.8	0.8	0.8
18	0.8	0.8	0.8	0.8	0.8	0.8

TAB. 6.6 – Taux de reconnaissance d'expression pour chaque image de la base, modifiée par l'algorithme d'anonymisation par **factorisation**, quelque soit l'identité cible. Le taux global de reconnaissance est de **69%**.

Expression \ Identité	**1**	**2**	**3**	**4**	**5**	**6**
1	0.7	0.7	0.0	0.0	0.3	0.8
2	0.7	0.7	0.2	0.2	0.7	0.7
3	0.3	0.3	0.3	0.3	0.3	0.7
4	0.5	0.5	0.5	0.5	0.5	0.5
5	0.3	0.7	0.3	0.7	0.2	0.7
6	0.5	0.5	0.3	0.5	0.3	0.7
7	0.8	0.8	0.7	0.8	0.7	0.8
8	0.7	0.8	0.7	0.7	0.3	0.7
9	1.0	1.0	0.2	0.2	0.8	1.0
10	0.3	0.7	0.2	0.3	0.3	1.0
11	1.0	1.0	0.5	0.5	0.7	1.0
12	0.5	0.7	0.3	0.7	0.2	0.7
13	0.8	0.5	0.2	0.7	0.8	1.0
14	0.3	0.8	0.3	0.7	0.8	0.7
15	0.7	0.7	1.0	0.8	0.7	0.8
16	0.3	0.3	0.2	0.5	0.3	0.7
17	0.7	0.7	0.8	0.5	0.5	0.8
18	0.7	0.7	0.7	0.7	0.7	0.8

TAB. 6.7 – Taux de reconnaissance d'expression pour chaque image de la base, modifiée par l'algorithme d'anonymisation par **translation**, quelque soit l'identité cible. Le taux global de reconnaissance est de **58%**.

anonymisées sont pertinents. L'anonymisation sera alors jugée uniquement sur sa capacité à modifier l'identité.

Le taux global de reconnaissance, calculé sur les 860 images de la séquence vidéo, est de **97.6%** lorsque l'identité a été modifiée par la méthode à base de projection, de **96.9%** pour la méthode par factorisation et de **95.7%** pour la méthode par translation.

Nous en profitons pour visualiser le résultat des trois méthodes d'anonymisation pour quelques images représentatives de l'ensemble de la séquence vidéo (voir la figure 6.14).

6.2.6 Qualité du rendu

Les expérimentations précédentes montrent que les deux méthodes par factorisation ou par projection ont des résultats similaires. La méthode par translation, en revanche, donne les plus mauvais résultats ; nous l'écartons donc.

On peut cependant remarquer que le rendu visuel n'est pas toujours très bon. Ceci vient de la faible résolution qui a été retenue pour la génération de la texture. En effet, la texture a été échantillonnée sur la forme moyenne $\mathbf{s}_0$ de dimension 48×48, la résolution ayant un effet direct sur les performances des algorithmes.

Les textures sont toutes échantillonnées sur une même forme référence $\mathbf{s}_0$ de manière à construire un espace vectoriel de textures. Lors de la construction d'un AAM, cette forme référence est classiquement la forme moyenne $\mathbf{s}_0$. Cependant, rien n'empêche pour les algorithmes d'anonymisation d'utiliser une forme référence différente de la forme moyenne $\mathbf{s}_0$. En particulier, il est possible de choisir une forme de référence où la bouche est ouverte.

De la même manière, la qualité du rendu peut être encore théoriquement améliorée pour l'algorithme d'anonymisation par projection. En effet, celui-ci renvoie pour chaque image un vecteur expressif $\mathbf{b}$ qui traduit le mélange des différentes expressions unitaires par combinaison linéaire. Il est donc possible, pour le rendu, de mélanger directement les images des visages de la base d'apprentissage, sans passer par un échantillonnage sur une forme de référence. L'échantillonnage est alors considéré idéal dans ce cas.

Cette manipulation n'est en revanche pas possible pour l'algorithme par factorisation. En effet, le vecteur expressif $\mathbf{b}$ ne correspond pas dans ce cas à des coefficients de mélanges entre les différentes expressions unitaires, mais à des coefficients appliqués à une base calculée $\mathbf{A}^i$ qui ne représente pas directement les images expressives de la base d'apprentissage.

La figure 6.15 présente des résultats d'anonymisation par projection, sur la même séquence que précédemment, avec un échantillonnage idéal de la texture et avec une forme de référence $\mathbf{s}_0$ échantillonnée à une résolution de 180×240 pixels, ce qui représente la boîte englobante maximale dans les images d'origine. On notera que dans ces deux cas, la qualité visuelle du rendu est

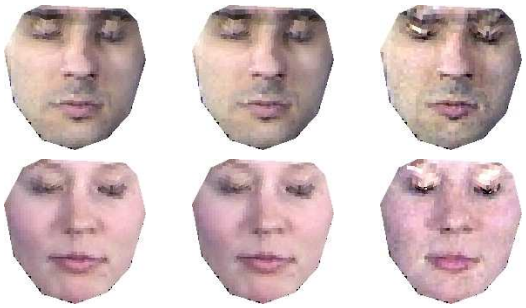
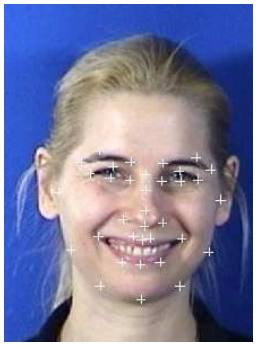
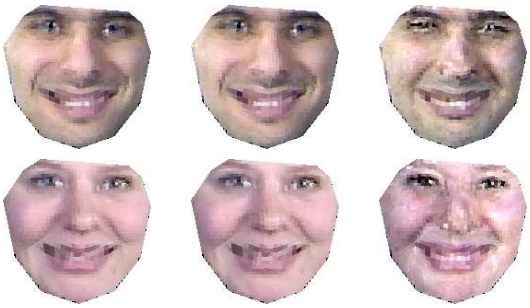
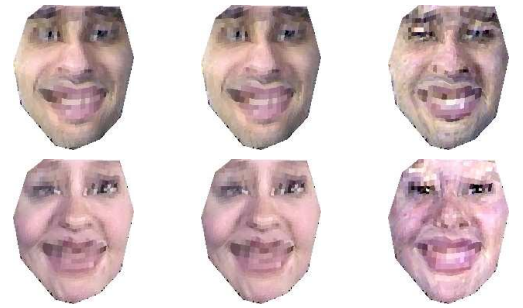
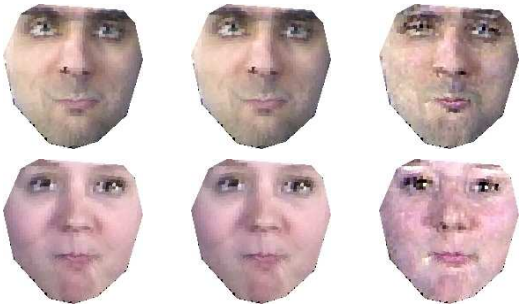
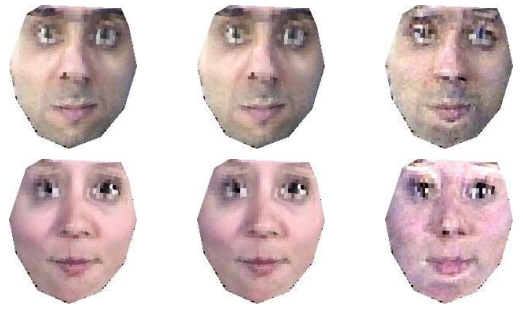
Image	Anonymisation par		
	Projection	Factorisation	Translation
 11			
 45			
 51			
 115			

Image	Anonymisation par		
	Projection	Factorisation	Translation
 216			
 298			
 410			
 668			

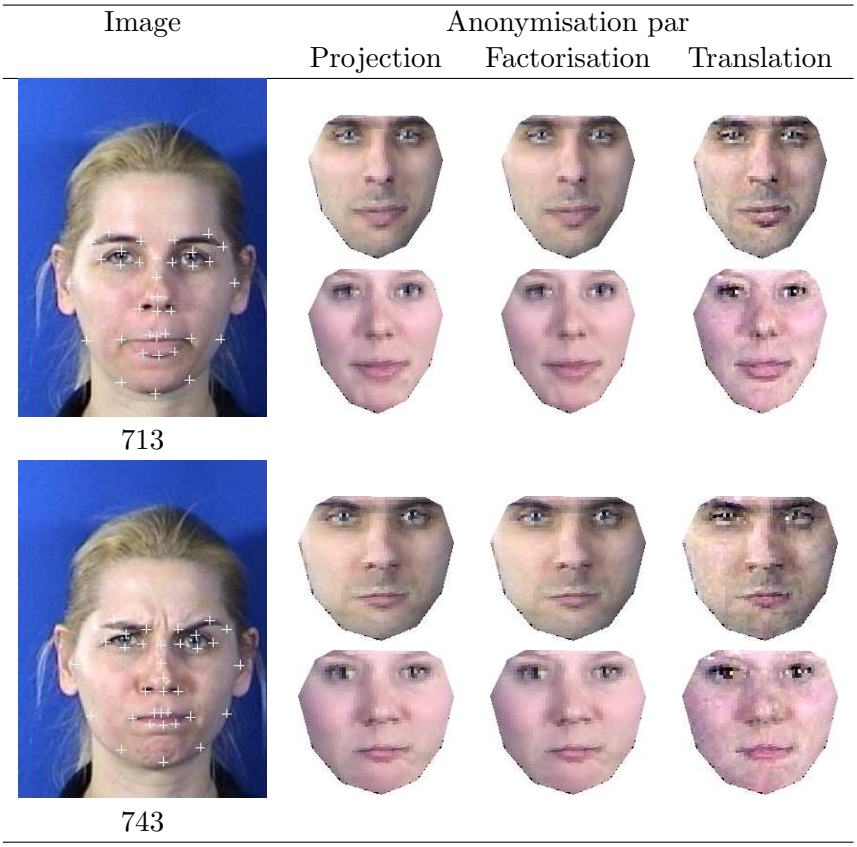


FIG. 6.14 – Résultats des algorithmes d'anonymisation pour les identités cibles 2 et 4 sur quelques extraits d'une séquence vidéo. La colonne de gauche donne le résultat de l'algorithme d'adaptation d'AAM.

comparable.

Le bilan de ces évaluations est que les méthodes de changement d'identité par projection et par factorisation sont celles qui donnent les meilleurs résultats. Entre les deux, le fait que la méthode par projection permette l'extraction d'un vecteur expressif pouvant être « expliqué » comme étant une combinaison d'expressions unitaires dont les images existent dans la base d'apprentissage est un avantage. Ainsi, le vecteur expressif extrait a un sens qui peut être exploité, et en particulier permettre dans le cas de l'anonymisation d'avoir une qualité optimale de rendu.

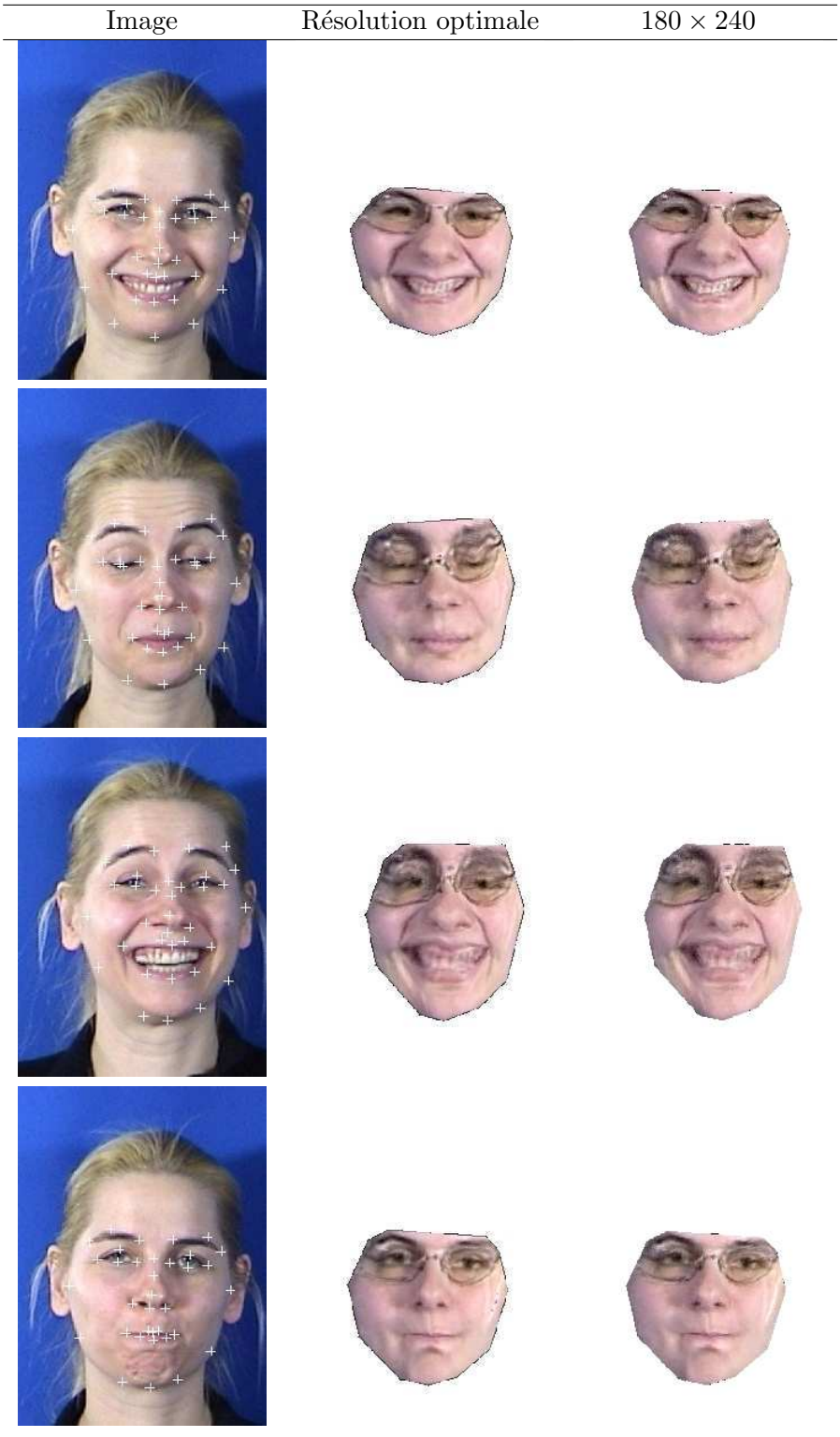


FIG. 6.15 – Exemples de rendu après anonymisation par la méthode par projection lorsque la résolution d'échantillonnage est optimale et pour une résolution de 180×240.

Chapitre 7

Conclusion et perspectives

Le travail présenté ici s'inscrit dans le domaine de l'analyse automatique des expressions faciales issues d'une captation vidéo de productions en langue des signes.

Nous avons présenté dans un premier temps le rôle des expressions faciales en langue des signes ainsi que des éléments d'anatomie faciale humaine.

Un état de l'art sur le thème de la description des expressions a été proposé, détaillant les formalismes de description et d'annotations manuels, utilisés par les linguistes et les psychologues ainsi que les formalismes informatisés. Puis nous avons distingué deux types de méthodes utilisées pour l'analyse automatique des déformations faciales : les méthodes basées sur une segmentation en composantes du visage et les méthodes globales n'utilisant pas de segmentation. Les méthodes à base de modèles déformables (à forme active ou apparence active) ont ensuite été détaillées.

Dans une deuxième partie, les modèles à apparence active et les algorithmes dits « à composition inverse » ont été présentés en détail. Ces algorithmes permettent le suivi de points d'intérêt du visage. Ils ont ensuite été évalués, notamment en terme de précision permettant de conclure qu'ils peuvent atteindre une grande précision de la localisation des points d'intérêt, à condition que la base d'apprentissage soit bien adaptée au problème, *i.e.* que le visage dont on cherche les déformations fasse partie de la base d'apprentissage du modèle déformable.

Nous avons étendu l'un de ces algorithmes de manière à permettre un suivi alors qu'une partie du visage observé est occulté, comme c'est le cas fréquemment en langue des signes. L'amélioration présentée a permis une amélioration de la détection automatique des occultations manuelles, par rapport aux travaux existants, et un suivi robuste de séquences vidéo de plusieurs centaines d'images.

Nous avons enfin détaillé différentes applications qui pouvaient être tirées de ces algorithmes, en présentant dans un premier temps une méthode de description qui considère une expression comme étant la combinaison de déformations faciales unitaires. Une application originale et spécifique au contexte de

la langue des signes a été présentée, permettant de rendre anonyme, par traitement vidéo, un enregistrement vidéo en remplaçant l'identité d'une personne sans pour autant dégrader ses expressions.

Rotations hors-plan

En considérant les objectifs initiaux, qui consistaient à permettre le suivi d'expressions faciales dans le contexte de la langue des signes, certains aspects n'ont pas été achevés. Il s'agit notamment de l'extraction des déformations faciales lorsque le visage est en rotation hors-plan.

Dans ce cas, il nous semble nécessaire d'étendre les méthodes utilisées au cas 3D. Il est possible, par exemple, de reconstruire le modèle 3D du visage (avec ses déformations 3D) à partir du résultat du suivi 2D sur un ensemble d'images, par des techniques de *structure from motion* (non rigide dans ce cas, voir B.1, page 142). Ces techniques ne sont cependant pas aisées à implanter et ce sujet est hors du contexte d'étude de cette thèse. C'est pourquoi nous ne les avons pas testé.

On peut toutefois dès maintenant envisager un suivi des déformations faciales sur une séquence contenant de faibles rotations hors-plan. Dans ce cas, le nombre de vecteurs de déformation et de variation de texture à retenir n'est pas trop important et l'algorithme convergera plus facilement. Il est possible d'envisager des corpus d'étude de la langue où les rotations hors-plan du visage sont faibles (dans un discours sans transfert personnel), alors qu'il est beaucoup plus difficile d'en envisager sans occultation par exemple. Avec les outils développés ici, il est donc possible de suivre les déformations faciales sur une séquence choisie de langue des signes réaliste, bien qu'ils ne puissent pas être utilisés pour le suivi de productions signées quelconques.

Pouvoir de généralisation

Le modèle à apparence active est construit à partir d'une base d'apprentissage et nous avons noté que cette base influe grandement sur les performances des algorithmes d'adaptation. Nous avons notamment observé que le visage dont on cherche les déformations doit faire partie de la base d'apprentissage pour obtenir une précision satisfaisante. De plus, les performances décroissent également quand trop de vecteurs de variation de formes sont retenus pour la construction du modèle, même s'ils sont issus d'une base ne contenant que l'identité de la personne étudiée. Ainsi, si l'on s'intéresse au suivi des déformations faciales observées sur une vidéo, il s'agit de construire une base d'apprentissage par mise en correspondance manuelle des points d'intérêt sur un ensemble d'images extraites de la vidéo. En dehors du fait que la mise en correspondance manuelle est une tâche pénible à effectuer, la sélection des images de la vidéo pour la construction d'une base d'apprentissage « optimale » demande une certaine expérience.

Une distinction doit être faite entre le pouvoir de représentation de l'AAM et les performances atteignables par un algorithme d'adaptation d'AAM. Un AAM est classiquement représenté par une base vectorielle de vecteurs de variation de forme et de texture. Plus ces bases contiennent de vecteurs, plus elles sont capables de représenter une donnée de l'espace original (espace des coordonnées 2D et des images) sans erreur. Les algorithmes d'adaptation effectuent une recherche de paramètres dans les espaces vectoriels engendrés par ces bases. Or, plus les bases contiennent de vecteurs, plus les espaces vectoriels représentent au mieux des visages, mais sont également capables de représenter des données qui n'ont rien à voir avec des visages.

Il est possible de restreindre la taille de l'espace de recherche, en imposant certaines contraintes, notamment en supposant que les données de la base d'apprentissage forment une ellipsoïde. Cependant cette modélisation n'est pas suffisante si la base d'apprentissage représente des variations de différents types (identités, expressions, poses, types d'éclairage). Une modélisation plus fine consisterait à considérer les données d'apprentissage comme faisant partie de plusieurs classes distinctes (par un algorithme de segmentation type $k-$moyennes) ou bien encore comme étant décrites par un mélanges de gaussiennes. Un telle modélisation permettrait de mieux coller aux données. Le problème classique du compromis biais-variance se posant cependant lorsque l'on chercherait à représenter un visage n'appartenant pas à la base d'apprentissage. Un tel espace de recherche peut être vu comme étant la combinaison de plusieurs espaces de plus petite taille, plus « locaux ». Si l'algorithme d'adaptation d'AAM n'est utilisé, à un instant donné, que sur un des espaces locaux, avec une base de déformations de forme et de texture spécifique à ce sous-espace, la rigidité du modèle serait plus grande et sa propension à tomber dans des minima locaux plus faible, à supposer que l'on dispose d'une stratégie efficace de basculement d'un espace local à l'autre (en utilisant une distance de Mahalanobis à chacun des sous-espaces par exemple). Et si le temps de convergence n'est pas un problème, rien n'empêche d'utiliser des techniques d'optimisation qui tendent vers un optimum global (des méthodes stochastiques notamment, telle que le recuit simulé [Kirkpatrick 83] par exemple).

Une telle approche permettrait d'utiliser une base d'apprentissage de taille arbitraire sans perturber les performances des algorithmes d'adaptation d'AAM. Ainsi le problème de généralisation à un visage inconnu pourrait être abordé.

Évaluation du modèle linéaire

Le modèle linéaire d'extraction des intensités d'activation d'expression, présenté en 6.1 a été évalué indirectement par le résultat de la méthode d'anonymisation par projection. Il mériterait cependant d'être évalué plus en détails.

Le problème vient du fait qu'il est difficile d'obtenir une vérité terrain, donnant à chaque instant l'intensité d'activation de chaque déformation unitaire.

L'évaluation peut néanmoins être faite avec une application d'animation faciale. En effet, en supposant un modèle 3D photo-réaliste, déformable en

expressions, à notre disposition, il est possible alors de générer des images qui serviront à la définition du modèle linéaire des intensités d'activation : il s'agirait de l'image du modèle 3D en expression neutre, et des images du modèle 3D synthétisées au pic de chacune des déformations unitaires. Les modèles de forme 2D sur chacune des images seraient obtenus directement par une projection 2D des coordonnées 3D du modèle de visage, ainsi aucune erreur ne serait introduite par une annotation manuelle. Ce protocole permettrait alors de tester l'efficacité du modèle linéaire. Reste cependant à supposer que les images du modèle 3D sont suffisamment photo-réalistes pour que les résultats puissent être extrapolés à une image réelle.

Évaluations qualitatives

Les différentes techniques présentées dans cette thèse – suivi des déformations faciales, description des expressions et anonymisation – mériteraient d'être évaluées de manière qualitative.

En effet, le contexte particulier d'étude (la langue des signes française) est un contexte langagier. Il est donc possible d'associer à une séquence vidéo une mesure d'intelligibilité par un panel d'usagers de la langue, la finalité étant que le message initial ne soit pas dégradé.

L'évaluation peut être menée à plusieurs niveaux dans la chaîne de traitement :

- en amont, après application de l'algorithme de suivi des déformations faciales. Dans ce cas, le test peut s'effectuer sur des vidéos synthétisées à partir de ce qui est extrait par l'algorithme de suivi. En particulier, il serait intéressant de savoir si les séquences qui ne peuvent pas être reconstruites (quand l'algorithme n'a plus assez de données fiables pour suivre les expressions car les occultations deviennent trop fortes) ne dégradent pas la compréhension globale du message ;
- en aval, après traitement par anonymisation. Il s'agit dans ce cas de tester l'intelligibilité du discours sur un identité anonymisée.

Outre le fait que cette évaluation demande une somme importante de données, se posent les problèmes de conception du protocole expérimental et de définition de la mesure de compréhension, qui nous semble être des problèmes de recherche du domaine de la linguistique.

Application à l'étude de la langue des signes

La langue des signes étant une langue sans forme écrite propre, le moyen de communication privilégié est l'enregistrement vidéo. Ce médium est ainsi naturellement l'objet étudié lorsque l'on s'intéresse au fonctionnement de la langue des signes. Dans ce contexte, l'analyse est généralement effectuée par une description la plus complète possible de chaque image de l'enregistrement vidéo.

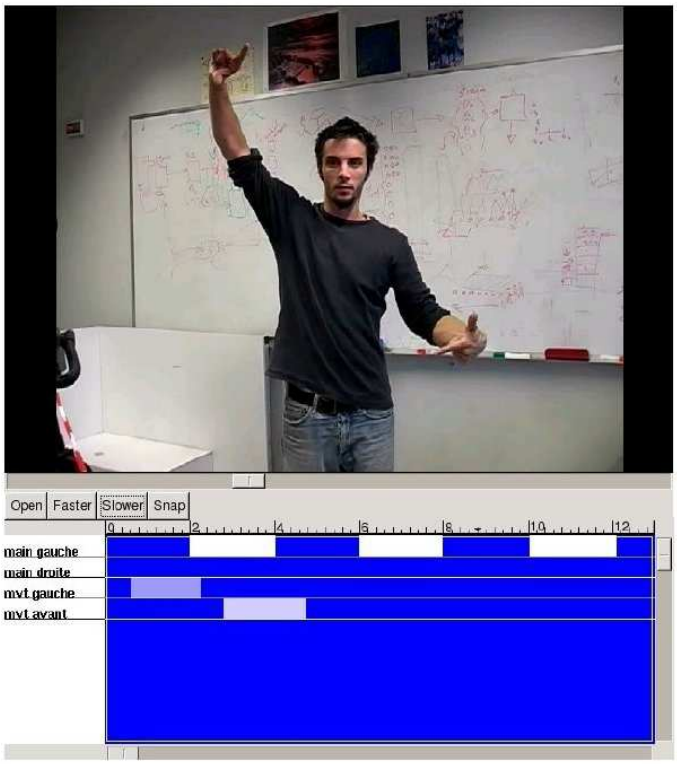


FIG. 7.1 – Éditeur en partition utilisé pour l'annotation de séquences vidéo [Braffort 04].

Concernant la description des expressions faciales, comme nous l'avons remarqué précédemment (voir en 2.2.3, page 26), la tâche est généralement peu aboutie. Ceci est sans doute dû au manque de formalismes simples de description des expressions faciales. C'est pourquoi une analyse assistée par ordinateur permettrait une description plus complète.

L'analyse de la langue des signes passe fréquemment par l'annotation d'un corpus d'enregistrement vidéo de productions signées. Bien que les annotations puissent se faire de manière linéaire en utilisant un formalisme tel qu'HamNoSys, il existe une autre approche qui consiste à découper l'étude en plusieurs paramètres et noter de manière temporelle l'évolution de chacun des paramètres. La granularité de découpage est laissée libre à l'opérateur et on trouve généralement un découpage selon les différents paramètres de définition d'un signe (emplacement, configuration de la main dominante, de la main dominée, expression faciale, etc.). Des outils informatiques ont été développés pour permettre l'annotation dans ce formalisme, appelé « en partition » (voir la figure 7.1).

Dans ce formalisme d'étude, la description des expressions pourrait correspondre à un chronogramme d'évolution de chacune des déformations faciales

unitaires, obtenu par l'application d'un algorithme d'adaptation d'AAM et d'un algorithme d'extraction des intensités d'activation musculaire.

Application à l'animation faciale

La méthode d'extraction de l'intensité des activations musculaires présentée en 6.1 (page 89) peut être appliquée pour l'animation du visage d'un personnage virtuel 3D. En effet, une méthode populaire d'animation faciale consiste à utiliser des techniques de morphage 3D (*3D morphing*) : l'infographiste définit une expression à son pic d'intensité par déplacement d'un ensemble de sommets du modèle 3D de base (neutre en expression) et associe un paramètre de contrôle (typiquement entre 0 et 1) à chaque expression, permettant ainsi, par interpolation linéaire des coordonnées 3D, d'obtenir une expression à différentes intensités. Cette technique est généralement préférée pour l'animation des éléments « mous » du corps tels que le visage, la peau et les vêtements, au lieu d'une animation par définition d'une chaîne articulaire telle qu'utilisée pour l'animation du squelette.

Un tel formalisme est en lien direct avec le formalisme de description présenté précédemment. En effet, la description des expressions consiste à renvoyer une intensité d'activation entre la valeur neutre et une déformation à son pic d'intensité. Ainsi, il semble naturel d'envisager une application d'animation faciale à partir d'une description des expressions obtenue à partir d'une vidéo.

Ainsi, les expressions retenues pour l'analyse doivent avoir leur équivalent défini par le modèle 3D. Si l'on souhaite décrire des expressions comme étant la combinaison de N déformations faciales unitaires, l'algorithme d'extraction des intensités d'activation doit être évalué avec une base d'apprentissage contenant les N déformations unitaires. De même, une bibliothèque des N déformations du modèle 3D doit exister.

Les intensités extraites de la vidéo peuvent alors être appliquées directement comme étant les paramètres d'interpolation de chaque paramètre de morphing du modèle 3D.

Système d'analyse de la LSF

Les méthodes présentées dans cette thèse, bien qu'appliquées au contexte de la langue des signes et étendues pour la détection automatique d'occultations, ne sont pas spécifiques à ce contexte. Elles ne dépendent que de la base d'apprentissage utilisée. Elles peuvent en particulier être appliquées à des contextes d'études plus contraints que celui de la langue des signes.

De plus, l'ajout de connaissances spécifiques au contexte de la langue des signes permettrait une amélioration de l'efficacité ou de la robustesse.

Par exemple, le critère de divergence, présenté en 4.1 pourrait prendre en compte une information sur les déformations maximales du visage humain, ou sur les combinaisons impossibles pour améliorer la détection.

En liant le système d'extraction des déformations faciales à un système de suivi du corps et des mains, il serait possible de savoir si le visage est partiellement occulté ou non (sans avoir besoin de savoir à quel endroit en particulier) et de choisir entre une version classique ou robuste de l'algorithme.

Le système d'extraction pourrait ainsi être conçu pour basculer entre plusieurs versions à efficacité, robustesse et précision différentes (le *project-out* et le simultané par exemple) en fonction de la demande, en le supposant inclu dans un système où une analyse linguistique serait menée.

Applications intéractives

Il nous semble enfin, que l'implantation logicielle des méthodes a une conséquence sur le développement de futures améliorations ou applications. Une implantation permettant une application de suivi proche du temps réel, avec un périphérique de capture vidéo type webcam permettrait de tester « à la volée » de nombreuses configurations limites.

Ceci nécessite néanmoins de vérifier que les différentes approximations faites pour obtenir un algorithme exécutable en temps réel restent valables dans notre cas. En particulier, la prise en compte des occultations repose sur une variante robuste de l'algorithme d'adaptation d'AAM (simultané pondéré) qui pondère l'influence de chacun des pixels de l'image d'erreur. L'introduction de cette carte des occultations nécessite une mise à jour de la matrice hessienne à chaque itération (voir A.1 page 135).

La mise à jour systématique doit être évitée afin de tendre vers une exécution en temps-réel. Il est possible, par exemple, de considérer la carte des occultations $Q(\mathbf{x})$ comme étant une constante en chacun des triangles du modèle de forme (voir [Baker 03b] section 4.4.2), accélérant ainsi la mise à jour de la matrice $\mathbf{H}$.

Annexe A

Détails de calculs

A.1 Dérivations

Nous détaillons ici la dérivation de la fonction d'erreur de l'algorithme simultané pondéré, présenté au chapitre 5.

La fonction à minimiser est :

$$\sum_{\mathbf{x}} Q(\mathbf{x})E(\mathbf{x})^2$$

avec :

$$E(\mathbf{x}) = \left[\mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(x) - I(W(\mathbf{x};\mathbf{p}^s))\right]$$

Il s'agit alors de minimiser itérativement (voir [Baker 03a] en 3.1) :

$$\sum_{\mathbf{x}} Q(\mathbf{x}) \left[\mathbf{t}_0(W(\mathbf{x};\Delta\mathbf{p}^s)) + \sum_{i=1}^{m}(\mathbf{p}_i^t + \Delta\mathbf{p}_i^t)\mathbf{t}_i(W(\mathbf{x};\Delta\mathbf{p}^s)) - I(W(\mathbf{x};\mathbf{p}^s))\right]^2$$

En effectuant un développement de Taylor du premier ordre de $W(\mathbf{x};\Delta\mathbf{p}^s)$ en $W(\mathbf{x};0)$, on a :

$$\sum_{\mathbf{x}} Q(\mathbf{x}) \left[\mathbf{t}_0(\mathbf{x}) + \sum_{i=1}^{m} \mathbf{p}_i^t \mathbf{t}_i(\mathbf{x}) - I(W(\mathbf{x};\mathbf{p}^t)) + (\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i)\frac{\partial W}{\partial \mathbf{p}^s}\Delta\mathbf{p}^s + \sum_{i=1}^{m} \Delta\mathbf{p}_i^t \mathbf{t}_i(\mathbf{x})\right]^2 \quad \text{(A.1)}$$

En notant :

$$G(\mathbf{x}) = \left[(\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i)\frac{\partial W}{\partial \mathbf{p}_1^s}, ..., (\nabla\mathbf{t}_0 + \sum_{i=1}^{m} \mathbf{p}_i^t \nabla\mathbf{t}_i)\frac{\partial W}{\partial \mathbf{p}_n^s}, \mathbf{t}_1(\mathbf{x}), ..., \mathbf{t}_m(\mathbf{x})\right]$$

L'équation A.1 devient :

$$\sum_{\mathbf{x}} Q(\mathbf{x}) \left[E(\mathbf{x}) - G(\mathbf{x})[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t]\right]^2$$

En dérivant, on obtient :

$$2\sum_{\mathbf{x}} Q(\mathbf{x})G^T(\mathbf{x})\left[E(\mathbf{x}) - G(\mathbf{x})[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t]\right]$$

Et en posant cette dernière équation égale à 0, on trouve :

$$[\Delta\mathbf{p}^s, \Delta\mathbf{p}^t] = \mathbf{H}^{-1}\sum_{\mathbf{x}} Q(\mathbf{x})G^T(\mathbf{x})E(\mathbf{x})$$

avec

$$\mathbf{H} = \sum_{\mathbf{x}} Q(\mathbf{x})G^T(\mathbf{x})G(\mathbf{x})$$

A.2 Analyse de Procrustes

L'analyse globale de Procrustes consiste à « aligner » un ensemble de formes 2D. L'alignement consiste à trouver pour chaque forme une translation, une mise à l'échelle et un angle de rotation qui minimisent la différence avec les autres (somme des distances point à point).

Il s'agit d'un algorithme itératif du type :

1. Choisir une forme comme l'estimation actuelle de la moyenne,
2. Aligner toutes les autres formes à l'estimation actuelle de la moyenne,
3. Calculer la nouvelle forme moyenne,
4. Itérer en 2 si la moyenne n'est pas stabilisée

Pour l'alignement « local » d'une forme $\mathbf{a}$ (de coordonnées $(\mathbf{a}_x^i, \mathbf{a}_y^i)$) à une autre forme $\mathbf{b}$ (de coordonnées $(\mathbf{b}_x^i, \mathbf{b}_y^i)$), on cherche les paramètres de translations (t_x, t_y), de rotation θ et de mise à l'échelle s. Ces paramètres sont obtenus par la procédure suivante (voir [Cootes 04]) :

On suppose la forme $\mathbf{a}$ centrée à l'origine (*i.e.*, $\sum_i \mathbf{a}_x^i = \sum_i \mathbf{a}_y^i = 0$).

On obtient donc les paramètres de la translation par :

$$t_x = \frac{1}{n}\sum_i \mathbf{b}_x^i$$

$$t_y = \frac{1}{n}\sum_i \mathbf{b}_y^i$$

On calcule ensuite :

$$\alpha = (\mathbf{a}^T\mathbf{b})/|\mathbf{a}|^2$$

$$\beta = \sum_i^n (\mathbf{a}_x^i\mathbf{b}_y^i - \mathbf{a}_y^i\mathbf{b}_x^i)/|\mathbf{a}|^2$$

On obtient alors :

$$s = \sqrt{\alpha^2 + \beta^2}$$

$$\theta = \text{atan}\left(\frac{\beta}{\alpha}\right)$$

A noter qu'une autre formulation, utilisant une décomposition en valeurs singulières (SVD) pour le calcul du paramètre de rotation existe (voir à ce propos [Akca 03]).

A.3 Analyse de texture

Chaque forme peut être vue comme un maillage de triangles, en utilisant une triangulation de Delaunay sur la forme de référence.

On considère que la transformation géométrique d'une forme vers la forme de référence peut être définie par une transformation affine en chacun des triangles. Le problème est alors le remplissage des pixels de la forme de référence, sachant que l'image aura subi une transformation affine en chaque triangle.

Le repère de la forme de référence est choisi en spécifiant le nombre de pixels à retenir pour la définition d'une texture. On définit par exemple une texture sur l'image $T(\mathbf{x})$ de taille 64×64.

L'algorithme de remplissage est le suivant :

Soit une forme $\mathbf{s}$ sur une image $I(\mathbf{x})$.
Soit $\mathbf{s} = \mathbf{s}_0 + \mathbf{Sp}^s$.
Pour chaque pixel $\mathbf{y}$ de $T(\mathbf{y})$ **Faire**
    $t \leftarrow triangle\_map(\mathbf{y})$
    $[\alpha, \beta] \leftarrow barycentric\_coord(\mathbf{y}, t, \mathbf{s}_0)$
    $[w_0, w_1, w_2] \leftarrow vertex\_coord(t, \mathbf{s})$
    $\mathbf{x} \leftarrow w_0 + \alpha(w_1 - w_0) + \beta(w_2 - w_0)$
    $T(\mathbf{y}) \leftarrow I(\mathbf{x})$
Fin Pour

La fonction *triangle_map* renvoie le numéro de triangle du maillage correspondant à la coordonnée passée en paramètre. Les coordonnées sont exprimées dans le repère de $T(\mathbf{x})$. Cette carte de correspondance peut être pré-calculée, puisque le remplissage de texture se fait toujours vers des coordonnées fixées.

La fonction $vertex\_coord(t, s)$ renvoie les coordonnées des trois sommets du triangle t de la forme s.

La fonction $barycentric\_coord(y, t, s)$ calcule les coordonnées barycentriques du point y dans le triangle t de la forme s. Ces coordonnées sont invariantes aux transformations affines, ce qui permet de calculer la transformée affine de chaque pixel d'un triangle en reportant les coordonnées barycentriques dans le triangle modifié. Si (v_0, v_1, v_2) représentent les coordonnées des sommets du

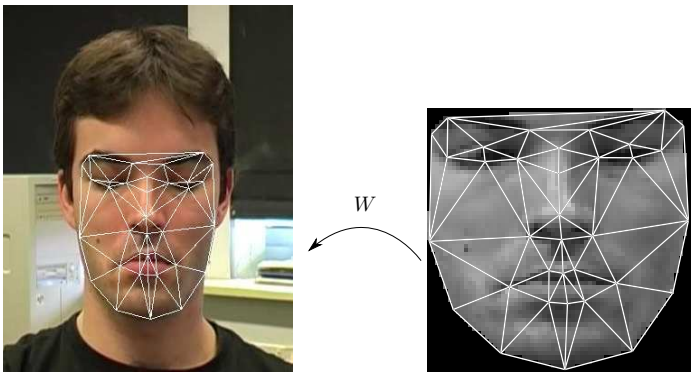


FIG. A.1 – Projection de la texture du visage sur la forme de référence.

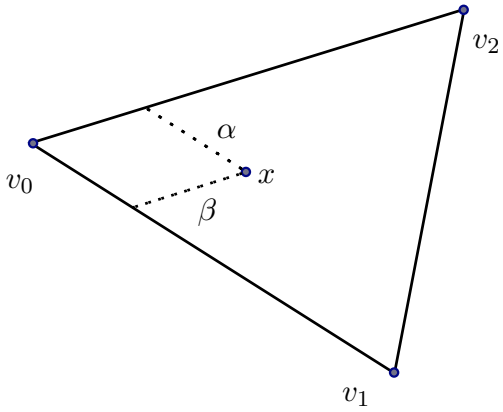


FIG. A.2 – Coordonnées barycentriques (α, β) du point x dans le triangle défini par les sommets v_0, v_1 et v_2

triangle t de la forme $\mathbf{s}_0$, la formule de calcul des coordonnées barycentriques α et β est :

$$\begin{bmatrix} \alpha(\mathbf{x}) \\ \beta(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} (v_1^x - v_0^x) & (v_2^x - v_0^x) \\ (v_1^y - v_0^y) & (v_2^y - v_0^y) \end{bmatrix}^{-1} \begin{bmatrix} x - v_0^x \\ y - v_0^y \end{bmatrix}$$

Soit :

$$\alpha(\mathbf{x}) = \frac{(x - v_0^x)(v_2^y - v_0^y) - (y - v_0^y)(v_2^x - v_0^x)}{(v_1^x - v_0^x)(v_2^y - v_0^y) - (v_2^x - v_0^x)(v_1^y - v_0^y)} \tag{A.2}$$

et

$$\beta(\mathbf{x}) = \frac{(y - v_0^y)(v_1^x - v_0^x) - (x - v_0^x)(v_1^y - v_0^y)}{(v_1^x - v_0^x)(v_2^y - v_0^y) - (v_2^x - v_0^x)(v_1^y - v_0^y)} \tag{A.3}$$

Ces coordonnées barycentriques sont ensuite reportées sur la forme $\mathbf{s}$, proportionnellement aux coordonnées de chacun des triangles (w_1, w_1, w_2).

Une fois la texture de tous les visages de la base d'apprentissage calculée par projection sur la forme moyenne, il est possible d'en faire une analyse statistique. De manière analogue à l'analyse statistique de forme, on effectue une analyse en composantes principales sur l'ensemble des textures (vectorisées).

Cependant, un problème peut se poser lors du calcul des vecteurs propres de la matrice de covariance. En effet, celle-ci est de taille $M \times M$ où M est le nombre de pixels de l'image de référence $T(\mathbf{x})$. Ce nombre est généralement élevé et peut amener à des calculs très coûteux, voire impossibles avec les capacités actuelles.

Pour remédier à ce problème, on procède à l'analyse statistique avec une version modifiée de l'analyse en composantes principales [Turk 91]. Cette méthode est valable lorsque la dimension des données est plus importante que le nombre d'échantillons dans l'ensemble d'apprentissage.

En reprenant les notations utilisées pour l'analyse de forme, on a, de manière générale, si v est vecteur propre de la matrice $B^T B$, associé à la valeur propre γ, alors

$$B^T B v = \gamma v$$

Et donc, en multipliant à gauche par B :

$$B B^T B v = \gamma B v$$

Ce qui indique que le vecteur Bv est vecteur propre de la matrice BB^T.

Ainsi, la première étape du calcul consiste à calculer les vecteurs propres de la matrice $B^T B$ qui est de dimension bien plus faible que BB^T. Les vecteurs propres de la matrice de covariance $C = BB^T$ seront obtenus en multipliant les vecteurs propres de $B^T B$ par B.

Annexe B

Prise en compte des rotations hors-plan

Les AAM sont définis par des modèles de forme à deux dimensions. Il est néanmoins possible de construire un modèle de forme qui prenne en compte les déformations dues aux rotations hors-plan d'un visage, qui seront alors considérées comme pouvant être expliquées par une combinaison linéaire de déformations 2D.

Le pouvoir de représentation du modèle est cependant encore une fois à distinguer de la difficulté de l'algorithme d'adaptation à converger. Bien que les déformations 3D puissent être expliquées par la statistique de forme, les vecteurs de déformation associés n'ont généralement que peu de sens. Autrement dit, les vecteurs de déformations qui permettent d'expliquer les rotations hors-plan (qui doivent être au minimum au nombre de six d'après [Xiao 04]) génèrent de nombreuses configurations non-réalistes. Il est possible de contraindre l'évolution des paramètres de forme pour qu'ils correspondent à des mouvements 3D réalistes [Xiao 04], mais il est nécessaire de disposer dans ce cas d'un modèle 3D du visage et de ses déformations.

De plus, les rotations hors-plan importantes génèrent des images où une partie du visage est cachée (par auto-occultation). Il est donc nécessaire d'utiliser une variante robuste de l'algorithme (comme présenté dans le chapitre suivant). Dans ce cas, une carte de confiance *a priori* peut être utilisée, calculée à partir d'une mesure d'orientation des triangles du modèle de forme (lors d'une forte rotation hors-plan, les triangles sont « retournés »).

Pour que les modèles de forme et de texture soient capables de prendre en compte des expressions affichées sur un visage en rotation hors-plan, la base d'apprentissage doit être très importante. En effet, à la différence d'un modèle 3D où la pose peut être exprimée indépendamment des expressions (ou de l'identité), ces deux dimensions sont corrélées dans le cas de modèles 2D.

B.1 Extraction de la pose 3D

Les rotations hors-plan d'un visage ne peuvent pas être prises en compte par un modèle linéaire, comme celui utilisé pour les déformations faciales. Autrement dit, la forme d'un visage tourné à 30 degrés ne peut pas être décrite comme la forme neutre à laquelle est additionnée la moitié de la déformation faisant passer de 0 à 60 degrés.

Bien qu'un modèle déformable 2D soit capable de suivre les rotations hors-plan du crâne (en retenant suffisamment de vecteurs de déformations 2D et en supposant que ce grand nombre de vecteurs n'affecte pas la performance de l'algorithme d'adaptation), l'information de pose 3D (angles de rotations) et de profondeur n'est pas accessible directement.

L'estimation de la pose 3D peut se faire en deux étapes : l'estimation de la forme 3D d'un modèle rigide de visage à partir d'un ensemble de coordonnées 2D puis l'estimation de la pose du modèle sur une nouvelle image.

Pour estimer la forme 3D d'un visage, il est possible d'utiliser des techniques de *structure from motion*. En particulier, en supposant un modèle de caméra à perspective faible, c'est à dire que les coordonnées 3D d'un objet (x, y, z) sont projetées dans le plan image par :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} q & 0 & 0 \\ 0 & q & 0 \end{bmatrix} \mathbf{R}(\rho, \theta, \phi) \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

où q est le facteur d'échelle et $\mathbf{R}(\rho, \theta, \phi)$ la matrice des rotations 3D de taille 2×3. Les vecteurs de translations n'apparaissent pas ici et on supposera que les objets ont été centrés.

De même si $\mathbf{s}_{3D}$ est une matrice $3 \times v$ de v points 3D et $\mathbf{s}_{2D}$ une matrice de v points 2D, alors :

$$\mathbf{s}_{2D} = \mathbf{QR}\mathbf{s}_{3D}$$

avec $\mathbf{Q}$ la matrice diagonale des facteurs d'échelle.

Si l'on dispose des coordonnées d'un objet $\mathbf{s}_{3D}$ projeté en 2D (en n'ayant subi que des rotations et un changement d'échelle) sur un ensemble de F images différentes, alors :

$$\mathbf{W} = \begin{bmatrix} \mathbf{s}_{2D}^1 \\ \vdots \\ \mathbf{s}_{2D}^F \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_1\mathbf{R}_1 \\ \vdots \\ \mathbf{Q}_F\mathbf{R}_F \end{bmatrix} \mathbf{s}_{3D}$$

et on pose :

$$\mathbf{M} = \begin{bmatrix} \mathbf{Q}_1\mathbf{R}_1 \\ \vdots \\ \mathbf{Q}_F\mathbf{R}_F \end{bmatrix}$$

et :

$$\mathbf{B} = \mathbf{s}_{3D}$$

avec $\mathbf{W} \in \mathbb{R}^{2F \times v}$, $\mathbf{M} \in \mathbb{R}^{2F \times 3}$ et $\mathbf{B} \in \mathbb{R}^{3 \times v}$

B.1.1 Reconstruction 3D

La matrice $\mathbf{B}$ peut alors être calculée par une SVD en retenant les trois colonnes associées aux trois plus fortes valeurs singulières (détails dans [Tomasi 92]). Cette décomposition permet d'écrire :

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

On identifie alors (par exemple) :

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{U}\sqrt{\mathbf{\Sigma}} \\ \hat{\mathbf{B}} &= \sqrt{\mathbf{\Sigma}}\mathbf{V}^T \end{aligned}$$

La factorisation par SVD n'est pas unique et on détermine $\mathbf{M}$ et $\mathbf{B}$ à une transformation linéaire près, modélisée par une matrice $\mathbf{G} \in \mathbb{R}^{3\times3}$.

$$\mathbf{MB} = (\hat{\mathbf{M}}\mathbf{G})(\mathbf{G}^{-1}\hat{\mathbf{B}})$$

La matrice $\mathbf{G}$ peut être déterminée en remarquant que la matrice $\mathbf{M}$ est constituée de matrices $\mathbf{Q}_i\mathbf{R}_i$, orthogonales et dont chaque ligne a la même norme. Un ensemble de contraintes peut donc être explicité sur $\mathbf{M}\mathbf{M}^T$, ce qui permet de calculer les coefficients de la matrice $\mathbf{G}$ (voir [Poelman 93] pour les détails).

Les matrices de rotation $\mathbf{R}_i$, contenues dans la matrice $\mathbf{M}$ peuvent être déterminées à une rotation près. Il est alors possible d'imposer que la première image de la séquence représente l'objet 3D sans rotation et sans mise à l'échelle.

Une dernière ambiguïté subsiste néanmoins sur le « signe » de la forme 3D. Ceci vient du fait que, par construction, il est impossible de différencier les deux configurations.

Cette méthode peut être appliquée sur un ensemble de points rigides du modèle de visage et ainsi déterminer leurs coordonnées 3D. Elle a l'avantage, via l'utilisation de la SVD, d'être robuste à l'imprécision qui pourrait être introduite dans la détermination des coordonnées 2D sur chaque image, ce qui est typiquement le cas avec l'utilisation d'AAM.

La technique est utilisable quand la matrice $\mathbf{W}$ est au moins de rang 3. Ce qui implique de retenir au moins 3 formes avec des rotations différentes, hypothèse tout à fait réaliste.

B.1.2 Redressement du modèle

L'information sur la pose 3D du modèle permet de « redresser », c'est à dire, transformer le modèle pour l'observer de face. Cependant, ce redressement ne peut s'effectuer que sur les points rigides du visage utilisés pour la reconstruction 3D. Les autres points non-rigides ne peuvent pas être redressés directement.

Dans ce cas, l'utilisation de techniques de détermination de la structure 3D à partir du mouvement considéré comme éventuellement non-rigide (*non-rigid structure from motion*) permet d'estimer les coordonnées 3D de chacun des points du modèle à chaque instant, en supposant un nombre suffisant d'images [Xiao 06].

Le principe de l'algorithme est le même que dans le cas rigide : considérer une matrice des observations $\mathbf{W}$ regroupant l'ensemble des coordonnées 2D et la décomposer en une matrice du mouvement $\mathbf{M}$ et une matrice de forme $\mathbf{B}$. Le principe est étendu ici en considérant une matrice $\mathbf{B}$ qui contient une forme 3D moyenne et un ensemble de vecteurs de déformations 3D. Pour assurer l'unicité de la décomposition, on ajoute aux contraintes métriques sur les matrices de rotation, des contraintes sur l'orthogonalité des vecteurs de déformations 3D.

Bibliographie

[Abboud 04] Bouchra Nakad ABBOUD. *Analyse d'expressions faciales par modèles d'apparence*. thèse de doctorat, Université Technologique de Compiègnes, 2004.

[Ahlberg 01a] Jörgen AHLBERG. Candide-3 - an updated parameterised face. Rapport Technique, Dept. of Electrical Engineering, Linköping University, Sweden, janvier 2001.

[Ahlberg 01b] Jörgen AHLBERG. *Using the Active Appearance Algorithm for Face and Facial Feature Tracking*. Dans 2nd International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Realtime Systems, Vancouver, BC, Canada, 2001.

[Akca 03] Devrim AKCA. Generalized Procrustes analysis and its applications in photogrammetry. Rapport Technique, Institute of Geodesy and Photogrammetry - Swiss Federal Institute of Technology - Zürick, juin 2003.

[Baker 03a] Simon BAKER, Ralph GROSS et Iain MATTHEWS. Lucas-Kanade 20 years on : A unifying framework : Part 3. Rapport Technique CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, novembre 2003.

[Baker 03b] Simon BAKER, Ralph GROSS, Iain MATTHEWS et Takahiro ISHIKAWA. Lucas-Kanade 20 years on : A unifying framework : Part 2. Rapport Technique CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, février 2003.

[Baker 04a] Simon BAKER, Ralph GROSS et Iain MATTHEWS. Lucas-Kanade 20 years on : A unifying framework : Part 4. Rapport Technique CMU-RI-TR-04-14, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, février 2004.

[Baker 04b] Simon BAKER et Iain MATTHEWS. *Lucas-Kanade 20 Years On : A Unifying Framework. International Journal of Computer Vision*, 56(3) :221 – 255, mars 2004.

[Baker 04c] Simon BAKER, Iain MATTHEWS et J. SCHNEIDER. *Automatic Construction of Active Appearance Models as an Image Coding Problem. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10) :1380–1384, octobre 2004.

[Baker 04d] Simon Baker, Raju Patil, Kong Man Cheung et Iain Matthews. Lucas-Kanade 20 years on : Part 5. Rapport Technique CMU-RI-TR-04-64, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, novembre 2004.

[Bébian 25] Roch Ambroise Bébian. Mimographie, ou essai d'écriture mimique, propre à régulariser le langage des sourds-muets. 1825.

[Braffort 04] Annelies Braffort, Annick Choisier, Christophe Collet, Patrice Dalle, Frédérick Gianni, Boris Lenseigne et Jérémie Segouat. *Toward an annotation software for video of Sign Language, including image processing tools & signing space modelling* . Dans 4th International Conference on Language Resources and Evaluation - LREC 2004 , Lisbonne, Portugal, 25/05/2004-30/05/2004, volume 1, pages 201–203. European Language Resources Association (ELRA), mai 2004.

[Cootes 92] T. Cootes et C. Taylor. *Active Shape Models – Smart Snakes.* Dans British Machine Vision Conference, pages 267–275, Leeds, 1992.

[Cootes 94] T.F. Cootes, C.J. Taylor, D.H. Cooper et J. Graham. *Active shape models - their training and application. CVGIP : Image Understanding*, 61 :38–59, 1994.

[Cootes 01] T. F. Cootes, G. J. Edwards et C. J. Taylor. *Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6) :681–685, 2001.

[Cootes 04] C. F. Cootes et C. J. Taylor. Statistical models of appearance for computer vision. Rapport Technique, University of Manchester, 2004.

[Costen 02] Nicholas Costen, Timothy F. Cootes, Gareth J. Edwards et Christopher J. Taylor. *Automatic extraction of the face identity-subspace. Image Vision Computing*, 20 :319–329, 2002.

[Darwin 01] Charles Darwin. L'expression des émotions chez l'homme et les animaux. Rivages Poche, 2001.

[Dedeoglu 06] Goksel Dedeoglu, Simon Baker et Takeo Kanade. *Resolution-Aware Fitting of Active Appearance Models to Low-Resolution Images.* Dans Proceedings of the 9th European Conference on Computer Vision, ECCV 2006, pages 83 – 97. Springer-Verlag, mai 2006.

[Ekman 69] Paul Ekman, E. Richard Sorenson et Wallace V. Friesen. *Pan-Cultural Elements in Facial Displays of Emotion. Science*, 164(3875) :86 – 88, avril 1969.

[Ekman 78] Paul Ekman et W. V. Friesen. Facial action coding system (FACS) : Manual. Palo Alto : Consulting Psychologists Press, New-York, 1978.

[Eveno 03] Nicolas Eveno. *Segmentation des lèvres par un modèle déformable analytique.* thèse de doctorat, Institut National Polytechnique de Grenoble, 2003.

[Fagertun 05] J. Fagertun et M. B. Stegmann. The IMM frontal face database. Rapport Technique, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2005.

[Fang 90] K. T. Fang et Y. T. Zhang. Generalized multivariate analysis. Springer, décembre 1990.

[Fasel 03] Beat Fasel et Juergen Luettin. *Automatic Facial Expression Analysis : A Survey. Pattern Recognition*, 36(1) :259–275, 2003.

[Gevers 97] T. Gevers et A.W.M. Smeulders. *Color Based Object Recognition. Lecture Notes in Computer Science*, 1310 :319–327, 1997.

[Gross 01] Ralph Gross, Jianbo Shi et Jeffrey Cohn. Quo vadis face recognition ? Rapport Technique CMU-RI-TR-01-17, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, juin 2001.

[Gross 05] Ralph Gross, Iain Matthews et Simon Baker. *Generic vs. person specific active appearance models. Image and Vision Computing*, 23(11) :1080–1093, novembre 2005.

[Hager 98] G. D. Hager et P. N. Belhumeur. *Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10) :1025 – 1039, 1998.

[Hammal 06] Zakia Hammal. *Segmentation des Traits du Visage, Analyse et Reconnaissance d'Expressions Faciales par le Modèle de Croyance Transférable.* thèse de doctorat, Université Joseph Fourier, Grenoble, juin 2006.

[Hjelmas 01] Erik Hjelmas et Boon Kee Low. *Face Detection : A Survey. Computer Vision and Image Understanding*, 83(3) :236–274, 2001.

[Kanade 73] Takeo Kanade. *Picture Processing System by Computer Complex and Recognition of Human Face.* thèse de doctorat, Department of Computer Science, Kyoto University, Kyoto, Japan, 1973.

[Kanade 00] Takeo Kanade, Jeffrey F. Cohn et Y. Tian. *Comprehensive Database for Facial Expression Analysis.* Dans The 4th IEEE International Conference on Automatic Face and Gesture Recognition (FG'00), pages 46 – 53, mars 2000.

[Kass 87] M. Kass, A. Witkin et D. Terzopoulos. *Snakes – Active Contour Models. International Journal of Computer Vision*, 1(4) :321–331, 1987.

[Kelly 70] M.D. Kelly. Visual identification of people by computer. Rapport Technique AI-130, Stanford AI Project, Stanford, CA, 1970.

[Kirkpatrick 83] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi. *Optimization by Simulated Annealing. Science*, 220(4598) :671–680, mai 1983.

[li Tian 01] Ying li Tian, Takeo Kanade et Jeffrey F. Cohn. *Recognizing Action Units for Facial Expression Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2) :97–115, 2001.

[Littlewort 06] G. Littlewort, M. Bartlett, I. Fasel, J. Susskind et J. Morvellan. *An automatic system for measuring facial expression in video. Image and Vision Computing*, 24(6) :615–625, 2006.

[Lucas 81] Bruce D. Lucas et Takeo Kanade. *An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI).* Dans Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81), pages 674–679, Vancouver, BC, Canada, April 1981.

[Lyons 98] Michael Lyons, Shigeru Akamatsu, Miyuki Kamachi et Jiro Gyoba. *Coding Facial Expressions with Gabor Wavelets.* Dans Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, pages 200–205, Nara, Japan, avril 1998.

[Martinez 98] A.M. Martinez et R. Benavente. The AR face database. Rapport Technique 24, Computer Vision Center - Universitat Autònoma de Barcelona, juin 1998.

[Martinez 02] Aleix M. Martinez. *Recognizing Imprecisely Localized, Partially Occluded, and Expression Variant Faces from a Single Sample per Class. IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6) :748–763, 2002.

[Matthews 03] Iain Matthews et Simon Baker. Active appearance models revisited. Rapport Technique CMU-RI-TR-03-02, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, avril 2003.

[Mercier 05] Hugo Mercier et Patrice Dalle. *Face analysis : identity vs. expressions.* Dans 2e Congrès de l'International Society for Gesture Studies (ISGS) : Interacting Bodies / Corps en interaction, Lyon, Ecole normale supérieure Lettres et Sciences humaines, Juin 2005.

[Mercier 06] Hugo Mercier, Julien Peyras et Patrice Dalle. *Toward an Efficient and Accurate AAM Fitting on Appearance Varying Faces* . Dans 7th International Conference on Automatic Face and Gesture Recognition - FG2006 , Southampton, Royaume-Uni, 10/04/2006-12/04/2006, pages 363–368. IEEE, 2006.

[Moriyama 06] Tsuyoshi Moriyama, Takeo Kanade, Jing Xiao et Jeffrey F. Cohn. *Meticulously Detailed Eye Region Model and Its Application to Analysis of Facial Images. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5) :738–752, 2006.

[MPEG Working Group on Visual 01] MPEG Working Group on Visual. International standard on coding of audio-visual objects, part 2 (visual), ISO/IEC 14496-2 :2001. 2001.

[Pantic 00] Maja Pantic et Leon J. M. Rothkrantz. *Automatic Analysis of Facial Expressions : The State of the Art. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1424–1445, 2000.

[Pantic 04] Maja Pantic et Leon Rothkrantz. *Facial Action Recognition for Facial Expression Analysis from Static Face Images. IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, 34(3) :1449–1461, June 2004.

[Pantic 05] M. Pantic, M. F. Valstar, R. Rademaker et L. Maat. *Web-based database for facial expression analysis.* Dans Proceddings of the IEEE

International Conference on Multimedia and Expo (ICME'05), pages 317–312, Amsterdam, Pays-Bas, juillet 2005.

[Poelman 93] Conrad Poelman et Takeo Kanade. A paraperspective factorization method for shape and motion recovery. Rapport Technique CMU-CS-93-219, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, December 1993.

[Régent 04] Muriel Régent. La mimique faciale dans les systèmes de transcription et les représentations graphiques de la langue des signes. 2004. Rapport de Maîtrise de l'Université Paris 8.

[Romdhani 04] Sami Romdhani, Volker Blanz, Curzio Basso et Thomas Vetter. Handbook of face recognition, Chapitre Morphable Models of Faces, pages 217–245. Springer-Verlag, 2004.

[Romdhani 05] Sami Romdhani. *Face Image Analysis using a Multiple Feature Fitting Strategy.* thèse de doctorat, University of Basel, Suisse, 2005.

[Sakai 69] T. Sakai, M. Nagao et S. Fujibayashi. *Line Extraction and Pattern Recognition in a Photograph. Pattern Recognition*, 1 :233–248, 1969.

[Scherer 82] Klaus R. Scherer et Paul Ekman, éditeurs. Handbook of methods in nonverbal behavior research. Cambridge University Press, 1982.

[Theobald 06] Barry-John Theobald, Iain Matthews et Simon Baker. *Evaluating Error Functions for Robust Active Appearance Models.* Dans Proceedings of the International Conference on Automatic Face and Gesture Recognition, pages 149 – 154, Southampton, avril 2006.

[Tomasi 92] C. Tomasi et Takeo Kanade. *Shape and Motion from Image Streams under Orthography : a Factorization Method. International Journal of Computer Vision*, 9(2) :137–154, novembre 1992.

[Turk 91] Matthew Turk et Alex Pentland. *Eigenfaces for recognition. Journal of Cognitive Neuroscience*, 3(1) :71–86, 1991.

[Viola 04] Paul Viola et Michael J. Jones. *Robust Real-Time Face Detection. International Journal of Computer Vision*, 57(2) :137–154, mai 2004.

[Wiskott 97] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger et Christopher von der Malsburg. *Face Recognition by Elastic Bunch Graph Matching. IEEE Transactions Pattern Anaysis and Machine Intelligence*, 19(7) :775–779, 1997.

[Xiao 04] Jing Xiao, Simon Baker, Iain Matthews et Takeo Kanade. *Real-Time Combined 2D+3D Active Appearance Models.* Dans Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages 535 – 542, Washington, DC, juin 2004.

[Xiao 06] Jing Xiao, Jinxiang Chai et Takeo Kanade. *A Closed-Form Solution to Non-Rigid Shape and Motion Recovery. International Journal of Computer Vision*, 67(2) :233–246, avril 2006.

[Yang 02] Ming-Hsuan Yang, David J. Kriegman et Narenda Ahuja. *Detecting Faces in Images : A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1) :34–58, janvier 2002.

[Zhang 98] Zhengyou Zhang. Feature-based facial expression recognition : Experiments with a multi-layer perceptron. Rapport Technique 3354, Institut National de Recherche en Informatique et en Automatique, février 1998.

[Zhao 00] Zhao, R. Chellappa, A. Rosenfeld et P. Phillips. Face recognition : A literature survey. Rapport Technique CAR-TR-948, Center for Automation Research, University of Maryland, 2000.