Méthodes d'apprentissage pour l'estimation de la pose de la tête dans des images monoculaires

Kévin Bailly

▶ **To cite this version:**

Kévin Bailly. Méthodes d'apprentissage pour l'estimation de la pose de la tête dans des images monoculaires. Interface homme-machine [cs.HC]. Université Pierre et Marie Curie - Paris VI, 2010. Français. NNT : . tel-00560836

HAL Id: tel-00560836

https://theses.hal.science/tel-00560836v1

Submitted on 30 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE DE DOCTORAT

de l'Université Pierre et Marie Curie – Paris 6

présentée par

Kevin BAILLY

pour l'obtention du grade de

Docteur de l'Université
Pierre et Marie Curie – Paris 6

Spécialité

Informatique et Image

Méthodes d'apprentissage pour l'estimation de la pose de la tête dans des images monoculaires

soutenue publiquement le 9 juillet 2010 devant le jury composé de

Rapporteurs	Pierre-Yves Coulon	Professeur des Universités	Gipsa-lab – INPG Grenoble
	Michel Dhome	Directeur de Recherche CNRS	LASMEA – UBP Clermont-Ferrand
Examinateurs	Liming Chen	Professeur des Universités	LIRIS – Ecole Centrale Lyon
	Matthieu Cord	Professeur des Universités	LIP6 – UPMC Paris
	Lionel Prevost	Maitre de conférence, HDR	ISIR – UPMC Paris
	Renaud Séguier	Professeur adjoint	IETR – Supélec
Directeur	Maurice Milgram	Professeur émérite	ISIR – UPMC Paris

If you think [...] that anything like a romance is preparing for you, reader, you never were more mistaken. Do you anticipate sentiment, and poetry, and reverie ? Do you expect passion, and stimulus, and melodrama ? Calm your expectations ; reduce them to a lowly standard. Something real, cool, and solid, lies before you ; something unromantic as Monday morning, when all who have work wake with the consciousness that they must rise and betake themselves thereto.

Charlotte Brontë : Prélude à *Shirley*

Table des matières

Table des figures

Liste des tableaux

Introduction

L'interprétation rapide et sans effort des mouvements de la tête est une faculté que l'homme a développé et qui constitue un pilier de la communication interpersonnelle. Par exemple, un individu engagé dans une discussion aura naturellement tendance à orienter son visage vers son interlocuteur. Certains mouvements traduisent également l'état émotionnel d'une personne : la peur peut induire un mouvement brusque de la tête en direction du danger potentiel, une personne honteuse ou mal à l'aise aura tendance à détourner son visage pour masquer ses émotions.

Les mouvements de tête jouent aussi un rôle conscient dans le processus de communication. Un hochement de tête signifiera, suivant sa direction, le consentement ou au contraire la désapprobation. La tête peut également se substituer au doigt pour désigner une zone d'intérêt.

L'analyse des mouvements de tête, aussi aisée soit-elle pour un humain, constitue encore un défi pour les systèmes de vision, comme en témoigne le récent état de l'art de Murphy-Chutorian et Trivedi [2009]. Ce domaine de recherche est d'autant plus actif qu'il constitue un maillon essentiel dans la chaîne de traitement de nombreuses applications. Les interfaces de communication entre l'homme et la machine s'orientent vers de nouveaux paradigmes. Les traditionnels claviers et souris font progressivement place à des interfaces plus intuitives et dématérialisées qui s'appuient sur le mouvement humain. Dans le domaine du jeu vidéo, l'entreprise Microsoft développe une interface de capture du mouvement entièrement orientée vision [1]. Dans les technologies d'assistance, des solutions permettent de commander le curseur d'un ordinateur par des mouvements de tête. En robotique, la connaissance de l'orientation des visages offre aux machines de nouvelles capacités d'analyse et d'interaction. Et la liste des domaines d'application est encore longue (*cf.* figure 1) : biométrie, réalité augmentée, vidéo-surveillance, sécurité routière, analyse comportementale pour le marketing, photo numérique...

Figure 1 – Quelques exemples d'applications d'estimation de la pose de la tête : analyse de la vigilance du conducteur, essais virtuels de lunettes par réalité augmentée, analyse du comportement par vidéo-surveillance

1. projet Natal : http://www.xbox.com/en-US/live/projectnatal/

Contexte

Cette recherche a été initiée dans le cadre de PILE, Programme International pour le Langage de l'Enfant [2]. PILE est un projet médical piloté par l'hôpital Necker Enfants Malades qui vise à améliorer le dépistage et le traitement précoce des troubles du langage. Cette étude s'appuie sur l'analyse des regards, des gestes et des productions vocales d'enfants en bas âge. Notre équipe, constituée de Ryad Benosman, Xavier Clady, Maurice Milgram et moi-même avait en charge la composante image du projet. Notre rôle était de proposer des méthodes pour estimer la position 3D des mains du bébé et analyser les contacts visuels entre le bébé et sa mère (c'est-à-dire déterminer lorsque le bébé regarde en direction du visage de sa mère). Dans ce contexte, nos premiers travaux ont porté sur la mise en œuvre d'un réseau calibré de huit caméras (*cf.* annexe B page 151), focalisées sur trois zones d'intérêt : la tête du bébé, ses mains et le visage de sa mère [Bailly *et al.*, 2006a]. Chaque zone est couverte par au moins deux caméras afin de replacer les différents éléments analysés dans un repère commun 3D (pour évaluer le contact visuel mère-bébé notamment). Nous avons également développé des méthodes *ad hoc* pour la localisation du visage et des mains du bébé [Bailly *et al.*, 2006b; Bailly, 2007].

Au cours de ces premières recherches, l'estimation de l'orientation de la tête, première étape d'un système d'analyse du regard, a été identifiée comme un verrou technologique et constitue l'objet de cette thèse.

Problématique

Le but de cette thèse est de concevoir des méthodes d'estimation automatique de la pose de la tête dans des images. Géométriquement parlant, cela consiste à définir un repère associé au visage et à estimer sa rotation par rapport à un repère global. Ce dernier pourra par exemple être associé au corps et défini par les plans sagittal, frontal et transversal. Le plus souvent, le repère choisi est indépendant du corps : il peut s'agir d'une position de référence du visage dans une séquence vidéo (la première image de face, par exemple). On pourra également choisir le repère de la caméra comme repère de référence. Dans le cas d'une représentation indépendante du corps, le système ne fera pas de différence entre un mouvement de la tête et un mouvement de la caméra. On trouve couramment une dernière représentation qui consiste à définir uniquement une direction dans l'espace à l'aide des angles de rotation pan et tilt (*cf.* figure 2).

Pour qu'un système d'estimation de pose soit performant, il devra être robuste aux nombreuses sources de variations qui peuvent affecter l'image d'un visage :

– Déformations géométriques : on identifie deux sources principales de déformations géométriques, les variations liées aux expressions faciales (variations intra-personnelles) et celles liées à la morphologie (variations interpersonnelles).

2. PILE : http://www.psynem.org/Pile/

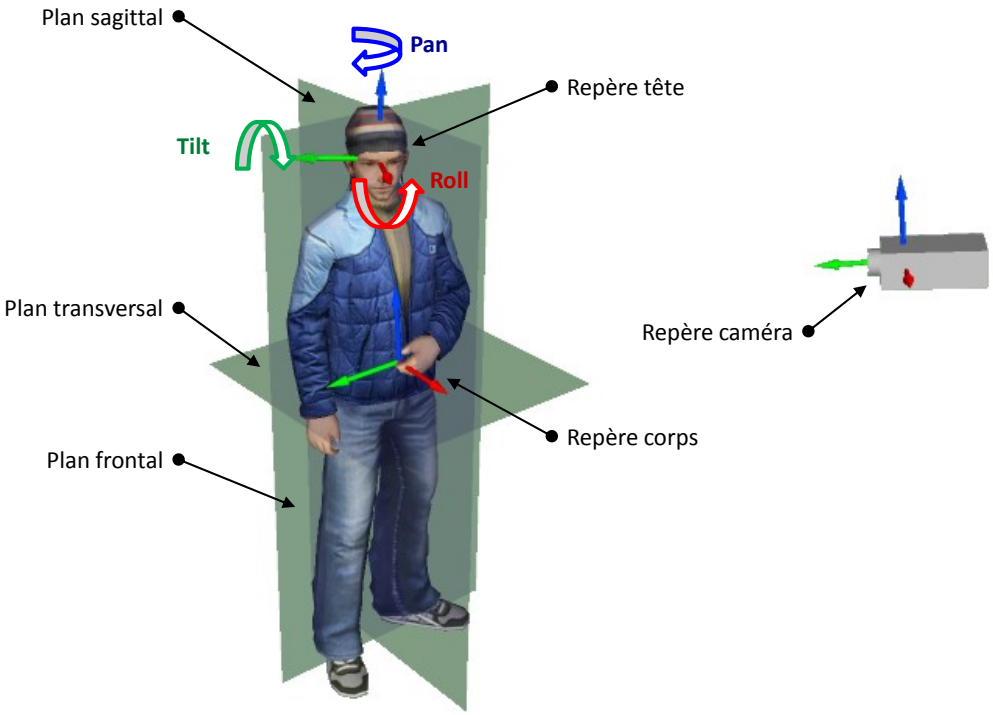


FIGURE 2 – Paramètres géométriques pour l'estimation de la pose de la tête

– Variations radiométriques : elles dépendent par exemple des propriétés de la peau (couleur, rides...), de la présence d'une barbe ou encore de la couleur des yeux. Si l'on fait une analogie avec la synthèse d'image, les variations radiométriques correspondent aux changements de texture.

– Occultations : certaines parties du visage sont cachées par des éléments extérieurs tels que des lunettes, un chapeau, ou une main. Il peut s'agir aussi d'auto-occultations liées à des rotations hors-plan de la tête.

– Changement d'illumination : les variations du type, de la position et de l'orientation des sources lumineuses ainsi que les propriétés de réflectance du visage agissent directement sur l'apparence du visage dans l'image.

– Variations liées au périphérique d'acquisition : la longueur focale, les distorsions optiques, la résolution et la compression de l'image sont autant de sources de variations qui auront un impact sur le rendu final de l'image.

Contributions

Les principales contributions apportées par ce travail de thèse sont de deux types. D'un point de vue méthodologique, nous avons proposé BISAR (*Boosted Input Selection Algorithm for Regression*), une méthode de sélection de caractéristiques adaptée aux problèmes

de régression qui se caractérise par :

- le critère fonctionnel flou (FFC, *Fuzzy Functional Criterion*), nouvelle mesure pour sélectionner des descripteurs images pertinents et
- une nouvelle stratégie pour sélectionner itérativement des entrées complémentaires d'un réseau de neurones.

D'un point de vue applicatif, cet algorithme a été validé au travers de deux méthodes d'estimation de la pose de la tête.

- Une approche globale : notre algorithme BISAR est utilisé pour apprendre directement la relation entre l'apparence d'un visage (ensemble des niveaux de gris qui correspondent au visage dans l'image) et sa pose.
- Une approche orientée modèle : on ajuste un modèle déformable sur une image pour localiser un ensemble de points caractéristiques qui seront ensuite utilisés pour calculer la pose.

Organisation du document

La suite de ce document s'articulera autour de deux parties qui coïncident avec nos contributions applicatives. Dans la *première partie*, nous aborderons donc les méthodes globales d'estimation de la pose.

Le *chapitre 1* propose un état de l'art des méthodes globales. Nous détaillerons l'étape de représentation des données d'une part et l'étape de décision d'autre part. Nous présentons également les jeux de données et les méthodes d'estimation de pose que nous utilisons pour évaluer et comparer notre approche.

Le *chapitre 2* introduit une méthode simple d'estimation de la pose par comparaison avec des images de synthèse. Plus spécifiquement, on aborde l'extraction d'un modèle texturé à partir d'une image et d'un ensemble de points identifiés sur le visage. On propose ensuite une mesure de ressemblance pour comparer une image réelle avec des instances de notre modèle de synthèse.

Le *chapitre 3* propose une méthode d'estimation de la pose par régression non lineaire. Ce chapitre présente tout d'abord un tour d'horizon des méthodes de sélection de descripteurs et des méthodes de boosting pour la régression. Il décrit ensuite notre algorithme d'apprentissage BISAR. Enfin, nous évaluons la pertinence de cette approche dans le cadre de l'estimation de la pose.

On quitte alors les approches globales pour s'intéresser, dans une *deuxième partie*, à l'estimation de la pose de la tête à l'aide de modèles déformables.

Le *chapitre 4* dresse un état de l'art des méthodes d'alignement de modèles dans des images. Nous décomposons et nous analysons ces méthodes suivant quatre axes : la modélisation de la forme, la modélisation de l'apparence, la fonction de coût et l'estimation des paramètres du modèle.

Le *chapitre 5* décrit notre méthode d'alignement suivant les quatre axes énoncés précédemment. Nous mettons l'accent sur la fonction de coût qui constitue le cœur de notre méthode. Cette fonction, apprise à l'aide de BISAR, évalue la qualité de l'alignement du modèle dans l'image. Nous testons notre approche sur diverses bases de données et comparons nos résultats avec une méthode récente de la littérature.

Le *chapitre 6* reprend notre méthode d'alignement et l'évalue dans le cadre d'une application d'estimation de la pose de la tête. Nous effectuons les tests sur des images de synthèse dont les paramètres de pose sont parfaitement connus et sur des images réelles provenant de diverses bases de données et d'Internet.

Enfin, nous concluons ce document et développons les perspectives ouvertes par ces travaux.

Etat de l'art

Pour améliorer la lisibilité et pour assurer une relative indépendance des chapitres, l'état de l'art à été réparti dans différentes sections de ce document. Nous listons ci-après leurs emplacements :

- Méthodes globales d'estimation de la pose de la tête : chapitre 1 page 19
- Processus de sélection de descripteurs : section 3.1 page 57
- Méthodes de *boosting* pour la régression : section 3.2 page 61
- Méthodes d'alignement de modèles déformables : chapitre 4 page 87
- Estimation de la pose à partir d'un modèle déformable : section 6.1 page 133

Première partie

Estimation de la pose de la tête par une méthode globale

Notations

Généralités

s	Scalaire
$\mathbf{v}$	Vecteur
$\mathbf{M}$	Matrice
N	Constante
$\|.\|$	Norme euclidienne

Notations relatives à la selection des descripteurs

$1 \leq k \leq N$	Indice des descripteurs
$1 \leq i \leq M$	Indice des exemples
$1 \leq t \leq T_{max}$	Indice des itérations

$\mathcal{A}$	Ensemble d'apprentissage labélisé, $A = \{(\mathbf{x_i}, y_i) \| i = 1 \dots M\}$
$\mathcal{V}$	Ensemble de validation
$\mathcal{T}$	Ensemble de test
$\mathcal{F}$	Ensemble des descripteurs potentiels, $\mathcal{F} = \{H_k \| k = 1 \dots N\}$
$\mathcal{FS}$	Ensemble des descripteurs selectionnés

$\mathbf{x_i}$	$i^{\text{ème}}$ exemple
y_i	Sortie désirée associée à $\mathbf{x_i}$
H_k	$k^{\text{ème}}$ descripteur de $\mathcal{F}$
$H^{(t)}$	Descripteur de $\mathcal{FS}$ sélectionné à l'itération t
$h_{k,i}$	Valeur du $k^{\text{ème}}$ descripteur calculée sur le $i^{\text{ème}}$ exemple $h_{k,i} = H_k(\mathbf{x_i})$

$\mathbf{w}^{(t)}$	Vecteur de poids sur les exemples [1] $\mathbf{w}^{(t)} = \left\{ w_i^{(t)} \| i = 1, \dots, M \right\}$
Reg_t	Regresseur à t entrées.
$\varepsilon^{(t)}$	Erreur d'estimation du régresseur Reg_t
$\varepsilon_i^{(t)}$	Erreur d'estimation associée au $i^{\text{ème}}$ exemple

.

1. Considérant l'ensemble $\mathcal{A}$ comme ordonné, la distribution des poids $w^{(t)}$ peut être vue comme un vecteur dont la composante $w_i^{(t)}$ est le poids de l'exemple numéro i à l'itération t

Chapitre 1

Etat de l'art

Dans cet état de l'art nous présentons les méthodes qui utilisent globalement l'image d'un visage pour en déduire sa pose. Nous les regroupons sous le terme de méthodes globales, par opposition aux méthodes par alignement d'un modèle déformable ou aux approches géométriques qui infèrent la pose à partir des positions relatives de certains éléments du visage tels que les yeux et la bouche. Les approches par alignement d'un modèle seront traitées dans la seconde partie du document. Le processus d'estimation par les méthodes globales peut se décomposer en quatre étapes successives (*cf.* figure 1.1) :

1. Prétraitements : le signal capturé par la caméra est prétraité pour simplifier les étapes suivantes, sans perdre l'information pertinente. Par exemple, une égalisation d'histogramme ajustera automatiquement les niveaux de gris ou les pixels de l'image seront seuillés afin d'éliminer l'arrière-plan de l'image.
2. Localisation du visage : cette étape extrait la zone de l'image contenant le visage dont on souhaite connaître la pose. Dans certaines méthodes, elle est intrinsèquement liée à l'étape d'estimation de pose.
3. Représentation des images : l'objectif est d'extraire les informations pertinentes et discriminantes contenues dans l'image en mesurant certaines propriétés (extraction de caractéristiques, section 1.1) ou en projetant les données dans un espace où la distribution des visages en fonction de leur pose est plus régulière. (section 1.2).
4. Décision : un algorithme d'apprentissage supervisé est utilisé pour estimer les paramètres de pose du visage. Nous avons différencié trois grandes approches. Les méthodes par comparaison avec des prototypes (section 1.3.1) déterminent la pose d'un visage en fonction du label des exemples (les prototypes) les plus ressemblants de la base d'apprentissage. Les méthodes par classification (section 1.3.2) regroupent les visages dont la pose est semblable et apprennent à déterminer le groupe auquel appartient un visage inconnu. Les méthodes par régression (section 1.3.3) apprennent la relation continue qui associe une pose à chaque image de visage.

Une étape facultative de post-traitements valide ou rejette la décision prise à l'étape précédente à partir d'informations supplémentaires liées au contexte. Dans une application de suivi par exemple, l'estimation de la pose pourra être filtrée en fonction des estimations précédentes.

Nous nous intéresserons plus particulièrement à l'étape de représentation des images (section 1.1 et 1.2) et d'estimation de la pose (section 1.3).

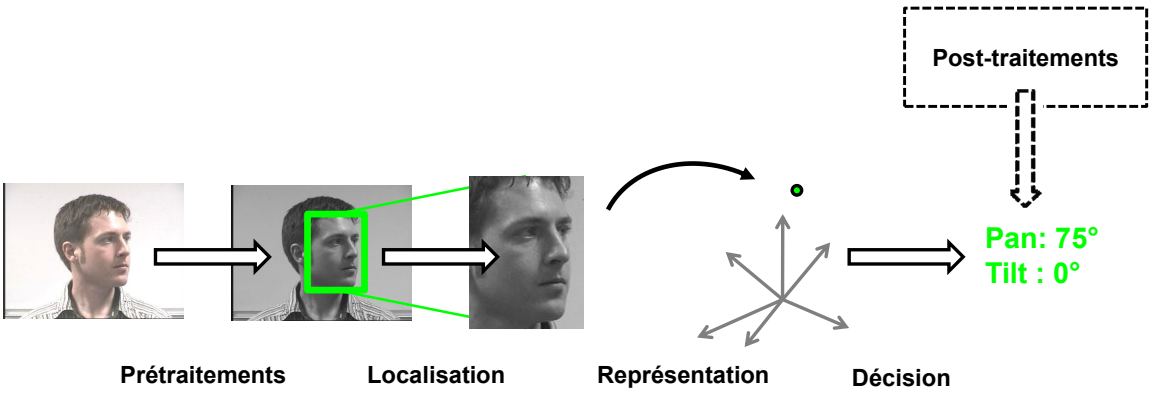


FIGURE 1.1 – Processus d'estimation de la pose par une méthode globale

1.1 Extraction de caractéristiques

L'objectif de cette étape est d'utiliser un ou plusieurs descripteurs pour représenter au mieux l'image en fonction de la tâche à réaliser. On cherchera généralement des descripteurs qui présentent des propriétés d'invariance par rapport aux caractéristiques non pertinentes. Pour un détecteur de véhicule par exemple, on s'intéressera à des descripteurs invariants aux variations colorimétriques puisque une voiture peut être de n'importe quelle couleur. Il existe de très nombreux descripteurs dans la littérature et l'objectif de cette section n'est pas de tous les énumérer. Nous présenterons les grandes catégories ainsi que les descripteurs les plus souvent utilisés.

1.1.1 Descripteurs globaux

Ils permettent d'extraire un ensemble d'attributs calculés sur toute l'image.

Filtrage de l'image Son but est d'extraire des informations pertinentes dans l'image. Les filtres de Sobel ou Canny, par exemple, sont utilisés pour extraire les contours qui présentent l'avantage d'être plus robustes vis-à-vis des changements d'illumination que les niveaux de gris. Les filtres de Gabor sont très largement utilisés en analyse d'image en général [Daugman, 1988] et en estimation de la pose de la tête en particulier [Wei *et al.*, 2002; Wu et Trivedi, 2008; Sherrah *et al.*, 2001]. Des opérateurs non linéaires tels que les *Local Binary Patterns* (LBP, Ojala *et al.* 1996) ont également donné de bons résultats en analyse de visages [Zhang *et al.*, 2007; Tan et Triggs, 2007]. Ma *et al.* [2006a] combinent les ondelettes de Gabor et les LBP pour estimer la pose de la tête.

Représentation par silhouette Elle consiste à binariser l'image de manière à séparer l'arrière-plan de la forme à analyser. La silhouette est ensuite caractérisée par un ensemble

de descripteurs de forme tels que les descripteurs de Fourier ou les moments géométriques [Mokhber *et al.*, 2008]. Zhang [2002] propose un panorama détaillé des descripteurs de forme. Les silhouettes issues de vues de différentes caméras peuvent également servir à reconstruire l'enveloppe visuelle de l'objet considéré. On ne cherche alors plus à caractériser les silhouettes directement mais l'objet 3D qui en résulte [Gond *et al.*, 2008].

Représentation par histogramme La représentation par histogramme est un outil simple et puissant qui a été utilisé dans de nombreux domaines. Les histogrammes de couleur sont invariants en translation et en rotation dans le plan image et varient lentement lors des rotations hors-plan, des changements d'échelles et des occultations. Les histogrammes donnent une signature compacte, stable et discriminante bien adaptée pour l'indexation de grandes bases de données [Swain et Ballard, 1991]. Les histogrammes de couleur sont sensibles à l'intensité et à la couleur de la source lumineuse ainsi qu'à la couleur de l'objet à détecter. Pour parer ce problème, Schiele et Crowley [2000] proposent par exemple de construire des histogrammes à champs récepteurs gaussiens multidimensionnels. L'histogramme de l'orientation des gradients pondérés par leur module (HoG de l'anglais *Histograms of Oriented Gradient*) est aussi très utilisé [Dalal et Triggs, 2005]. Pour conserver en partie l'information spatiale, on peut concaténer les histogrammes calculés dans des fenêtres glissantes ou disjointes de l'image [Lowe, 2004; Murphy-Chutorian *et al.*, 2007].

1.1.2 Descripteurs locaux

Par opposition aux descripteurs globaux, les descripteurs locaux ne caractérisent qu'une zone restreinte de l'image. Chaque descripteur extrait une information partielle et doit, par conséquent, être combiné à d'autres descripteurs pour fournir une représentation complète de l'image à analyser. Tuytelaars et Mikolajczyk [2008] proposent un tour d'horizon des descripteurs locaux tandis que Mikolajczyk et Schmid [2005] évaluent les performances de différents descripteurs tels que les *Shape Context* [Belongie *et al.*, 2001], les filtres orientables [Freeman et Adelson, 1991] ou SIFT [Lowe, 2004]. On peut distinguer deux manières de les utiliser :

1. On définit un ensemble de descripteurs locaux calculables en tout point de l'image avec différents paramètres. Cet ensemble dense et redondant doit être couplé avec une méthode de sélection de descripteurs. Viola et Jones [2004] combinent par exemple les descripteurs de Papageorgiou et Poggio [2000] inspirés des ondelettes de Haar (plus de 45̀000 descripteurs au total) avec l'algorithme AdaBoost [Freund et Schapire, 1997] qui sélectionne itérativement les meilleurs descripteurs.
2. On identifie des points d'intérêt dans l'image à l'aide de détecteurs spécifiques tels que [Harris et Stepheds, 1988] ou [Achard *et al.*, 2000], puis on utilise des descripteurs locaux (SIFT, Lowe 2004 ou SURF Bay *et al.* 2008, par exemple) pour caractériser le voisinage de ces points. Cette méthode offre une représentation compacte de l'image à analyser et ne nécessite pas d'étape de sélection des descripteurs. Il faut toutefois que la détection des points d'intérêt soit répétable, c'est-à-dire que les mêmes points

soient détectés au même endroit quelles que soient les conditions de prise de vue.

1.2 Réduction de dimension

Les images de visages sont des données de grande dimension. Il est toutefois raisonnable de supposer que ces données sont sur (ou à proximité d') une variété de faible dimension immergée dans l'espace ambiant $\mathbb{R}^N$ où N est le nombre de pixels (*cf.* figure 1.2). Les techniques qui cherchent une relation entre l'espace d'origine et un espace de faible dimension sont regroupées sous le terme de « réduction de dimension »ou plus récemment d'apprentissage de variétés (*manifold learning*). Dans notre cas, l'objectif est de trouver un espace de représentation sensible aux changements de poses et invariant aux autres sources de variation de l'image. Nous présentons dans cette partie un aperçu des méthodes de réduction de dimension, ainsi que leurs applications pour l'estimation de la pose de la tête. Pour une description plus exhaustive, le lecteur pourra se reporter à [Burges, 2005; Brun, 2007; van der Maaten *et al.*, 2009].

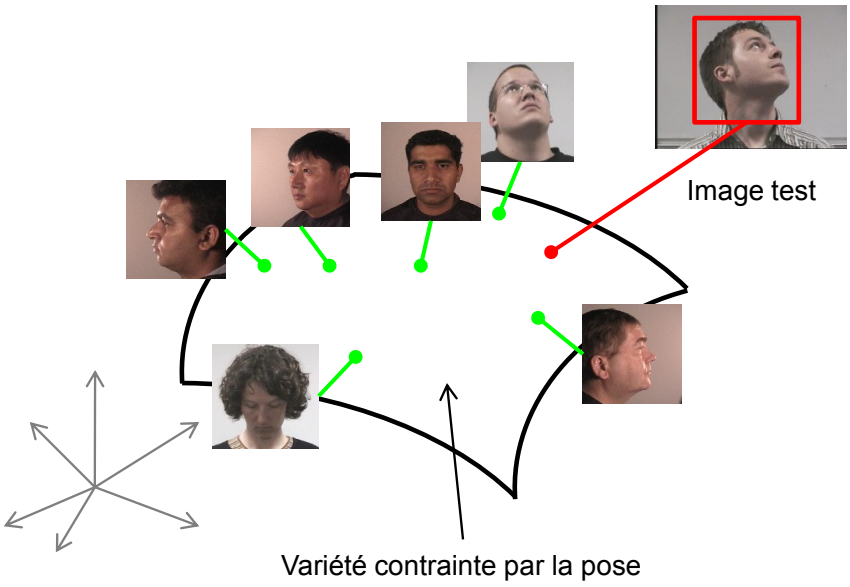


FIGURE 1.2 – Les techniques de réduction de dimension cherchent un sous-espace de projection de faible dimension qui représente au mieux les variations de pose du visage. Un visage dont la pose est inconnue sera projeté sur cet espace pour simplifier la prise de décision.

1.2.1 Méthodes non supervisées

Nous traitons dans un premier temps les méthodes dites non supervisées car elles ne prennent pas en considération la grandeur que l'on souhaite estimer (les paramètres de pose dans notre cas).

Méthodes linéaires L'Analyse en Composantes Principales (ACP) est une méthode de réduction de dimension linéaire introduite par Pearson [1901]. L'idée principale de l'ACP est de trouver une projection des données $\{\mathbf{x}_i\}_{i=1}^n \in \mathbb{R}^d$ qui maximise la variance. La base de ce nouveau sous-espace est formée des m premiers vecteurs propres de la matrice de covariance estimée,

$$\mathbf{C} = \frac{1}{n}\sum_i (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T \tag{1.1}$$

avec $\bar{\mathbf{x}}$, le vecteur moyenne estimée. Si on range les vecteurs propres dans une matrice $\mathbf{W}$ de taille $d \times m$, la projection $\mathbf{u}_i$ de chaque donnée $\mathbf{x}_i$ est calculée par

$$\mathbf{u}_i = \mathbf{W}^T(\mathbf{x}_i - \bar{\mathbf{x}}) \tag{1.2}$$

Cette méthode à été utilisée dans de nombreux domaines, en particulier en analyse faciale [Turk et Pentland, 1991]. Gong *et al.* [1996] ont montré qu'il suffisait de trois axes principaux pour représenter la distribution des visages en fonction de leur pose (*cf.* figure 1.3). Srinivasan et Boyer [2002] construisent un sous-espace propre par pose discrète.

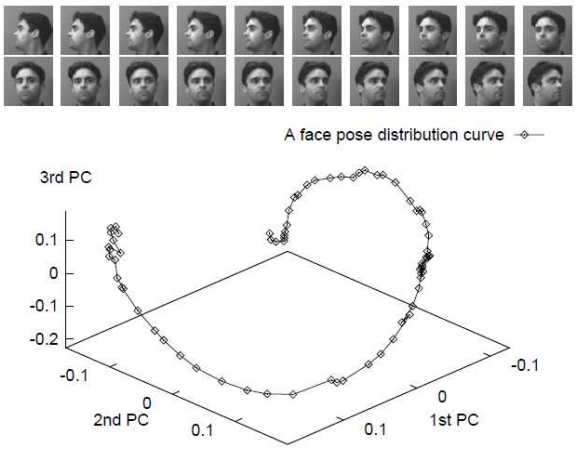


FIGURE 1.3 – Représentation des images d'un visage en rotation horizontale dans un espace formé par les 3 premiers vecteurs propres de l'ACP (tiré de Gong *et al.* [1996]).

Le *Multidimentional Scaling* (MDS) est une autre méthode très connue de réduction de dimension linéaire [Torgerson, 1952]. Au lieu de conserver la variance des données au cours de la projection, elle s'efforce de préserver toutes les distances entre chaque paire d'exemples $dist(\mathbf{x}_i, \mathbf{x}_j)$ en cherchant une transformation linéaire qui minimise l'énergie :

$$\varepsilon_{mds} = \sum_{i,j=1}^{n} \left(dist(\mathbf{x}_i, \mathbf{x}_j) - \|\mathbf{u}_i - \mathbf{u}_j\|\right)^2 \tag{1.3}$$

Ce problème de minimisation peut être résolu par une décomposition en valeurs propres [Cox et Cox, 2000; Williams, 2002]. Lorsque la fonction de distance entre les données est la distance euclidienne, les sorties $\mathbf{u}_i \in \mathbb{R}^m$ de la MDS sont les mêmes que les sorties de l'ACP. Elles sont obtenues par une rotation suivie d'une projection sur les axes de plus grande variance.

Méthodes par noyaux Les méthodes précédentes trouvent un espace de représentation fidèle aux données lorsque la structure de ces données est linéaire ce qui n'est généralement pas le cas. L'idée est d'utiliser une fonction noyau pour construire un espace de grande dimension dans lequel le problème devient linéaire. On peut ainsi appliquer des méthodes linéaires de réduction de dimension lorsque la structure intrinsèque des données n'est pas linéaire. Ces méthodes utilisent généralement « l'astuce du noyau » (de l'anglais *kernel trick*) qui établit que tout algorithme formulé avec une fonction noyau peut être reformulé avec une autre fonction noyau. Une démarche courante est d'exprimer la méthode avec un produit scalaire, puis de le remplacer par une fonction noyau. Le *kernel trick* permet alors de travailler dans l'espace transformé sans avoir à calculer explicitement l'image de chaque donnée. Wu et Trivedi [2008] utilisent par exemple une Analyse en Composante Principale avec Noyau (KPCA de l'anglais *Kernel Principal Component Analysis*, Schölkopf *et al.* 1998) pour représenter des visages dans différentes poses. Dans la KPCA, le choix de la fonction noyau n'est pas clairement défini. La MVU (de l'anglais *Maximum Variance Unfolding*) est une technique qui apprend conjointement la transformation et la fonction noyau [Weinberger *et al.*, 2004].

Méthodes par graphe PCA et MDS tentent de conserver les distances euclidiennes entre chaque paire d'exemples sans prendre en compte la distribution du voisinage de chaque exemple. Si les données reposent sur une variété très incurvée, deux points très proches au sens de la distance euclidienne peuvent être très éloignés si on considère la distance le long de la variété. Le jeu de données « brioche suisse » est couramment utilisé pour illustrer ce phénomène (*cf.* figure 1.4).

Depuis les années 2000 de nombreuses solutions ont été proposées pour réduire la dimension de l'espace en prenant en compte la topologie de l'ensemble de données. La méthode Isomap (pour *isometric feature mapping*, de Tenenbaum *et al.* 2000) par exemple, cherche une transformation qui préserve la distance géodésique entre les données. La distance géodésique est la distance la plus courte entre deux points, mesurée le long de la variété. L'algorithme se décompose en trois étapes :

1. Création d'un graphe d'adjacence G dans lequel chaque point $\mathbf{x}_i$ est connecté à ses k plus proches voisins (k-Isomap) ou aux données dont la distance est inférieure à ε (ε-Isomap).
2. Calcul du plus court chemin $d_G(\mathbf{x}_i, \mathbf{x}_j)$ entre chaque paire de nœuds du graphe à l'aide d'un algorithme du plus court chemin tel que [Dijkstra, 1959] ou [Floyd, 1962]. La matrice $\mathbf{D}_G = d_G(\mathbf{x}_i, \mathbf{x}_j)$ contient alors une estimation de la distance géodésique entre chaque point.
3. Construction d'une représentation de faible dimension par MDS appliquée à la matrice des distances $\mathbf{D}_G$.

Hu *et al.* [2005] utilisent Isomap pour découvrir la variété des visages dans différentes poses. La méthode Isomap présente toutefois quelques inconvénients. Elle est topologiquement instable [Balasubramanian et Schwartz, 2002] car elle peut créer des connexions

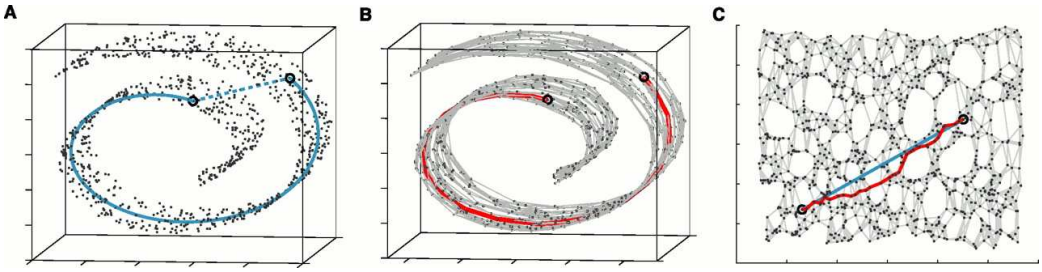


FIGURE 1.4 – Illustration de la méthode Isomap sur le jeu de données « brioche suisse » (tiré de Tenenbaum *et al.* 2000). (A) La distance entre deux exemples du jeu de données (pointillés) ne reflète pas leur ressemblance intrinsèque mesurée par la distance géodésique (trait plein). (B) Le graphe d'adjacence permet d'approximer (courbe rouge) la distance géodésique. (C) Projection des données dans l'espace 2D trouvée par Isomap.

fausses dans le graphe G ou mal approximer la distance géodésique lorsque la variété est trouée [Lee et Verleysen, 2005].

LLE (pour *Locally Linear Embedding*) est une technique de représentation de données de grande dimension dans un espace de faible dimension préservant *localement* la structure linéaire des exemples à proximité [Roweis et Saul, 2000]. En ne conservant que la structure locale des données, le problème se rapporte à la décomposition en valeurs propres d'une matrice creuse (par opposition aux méthodes précédentes qui devaient décomposer une matrice pleine). La première étape est semblable à Isomap puisqu'elle consiste à construire un graphe des k-plus proches voisins de $\mathbf{x}_i$. Chaque exemple est ensuite décrit par une combinaison linéaire de poids w_{ij} des k-plus proches voisins de $\mathbf{x}_i$. Les poids de reconstruction représentent la contribution du $j^{\text{ème}}$ point dans la reconstruction du $i^{\text{ème}}$ exemple. La dernière étape revient à trouver la représentation qui conserve ces relations entre exemples en minimisant la fonction de coût :

$$\varepsilon_{lle} = \sum_i \left\| \mathbf{u}_i - \sum_j w_{ij} \mathbf{u}_j \right\| \tag{1.4}$$

La matrice de covariance des sorties $\mathbf{y}_i$ doit être unitaire pour éviter les solutions triviales. Roweis et Saul [2000] ont montré que l'espace réduit qui minimise l'équation (1.4) est formé des m vecteurs propres associés aux m plus faibles valeurs propres non nulles de la matrice $(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$. $\mathbf{I}$ est la matrice identité $n \times n$ et $\mathbf{W}$ est la matrice creuse $n \times n$ dont les composantes sont égales aux poids de reconstruction si x_i est connecté à x_j et nulles sinon.

L'aspect local de LLE la rend moins sensible aux court-circuits que Isomap car seule une partie restreinte de la matrice $\mathbf{W}$ est affectée.

Il existe de nombreuses autres techniques par graphe. Certaines sont locales telles que les *Laplacian Eigenmaps* [Belkin et Niyogi, 2003] ou les *Local Tangent Space Alignment* [Zhang et Zha, 2005]. D'autres sont globales telles que les *diffusion maps* [Coifman et Lafon, 2006] ou le *Maximum Variance Unfolding* [Weinberger et Saul, 2006].

Le but de ces méthodes est de représenter chaque nœud du graphe par un vecteur de faible dimension qui préserve les similarités entre les paires de vecteurs. Toutefois ces techniques ne fournissent pas de projection explicite sur l'espace de faible dimension et il n'est par conséquent pas possible de projeter un nouvel exemple qui n'était pas présent dans l'ensemble de départ (*out of sample problem*). Plusieurs solutions ont été proposées pour résoudre ce problème [Raytchev *et al.*, 2004; Bengio *et al.*, 2003; Yan *et al.*, 2007]. Raytchev *et al.* [2004], par exemple, utilisent les *Locality Preserving Projections* [He *et al.*, 2003] qui définissent une relation pour n'importe quel point de l'espace.

L'autre inconvénient majeur des méthodes par graphe est que l'information de pose des visages n'est pas exploitée. Nous allons donc aborder les méthodes supervisées qui intègrent explicitement cette information dans le choix du sous-espace.

1.2.2 Méthodes supervisées

La méthode de réduction de dimension la plus connue et la plus souvent utilisée est certainement l'Analyse Discriminante Linéaire (LDA de l'anglais *Linear Discriminant Analysis*). L'objectif de la LDA est de trouver un sous-espace de projection qui maximise le ratio entre la matrice de covariance inter-classe $\mathbf{C}_{inter}$ et la matrice de covariance intra-classe $\mathbf{C}_{intra}$ (critère de Fisher). Cette optimisation se rapporte à la résolution d'un problème aux valeurs propres généralisé :

$$\mathbf{C}_{inter}\mathbf{w}_k = \lambda_k \mathbf{C}_{intra}\mathbf{w}_k \tag{1.5}$$

Les vecteurs propres associés aux plus grandes valeurs propres forment le sous-espace de projection. La LDA est une technique très efficace de réduction de dimension, mais elle se limite toutefois aux problèmes de classification. Kwak *et al.* [2008] proposent une extension de cette méthode aux problèmes de régression. L'idée est de considérer que les exemples $\mathbf{x}_i$ qui ont de faibles différences de label y_i appartiennent à la même classe. Les matrices de covariance sont remplacées par :

$$\mathbf{S}_{intra} = \frac{1}{n_{intra}} \sum_{i,j \in \mathcal{A}_{intra}} f(y_i - y_j)(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{1.6}$$

$$\mathbf{S}_{inter} = \frac{1}{n_{inter}} \sum_{i,j \in \mathcal{A}_{inter}} f(y_i - y_j)(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \tag{1.7}$$

avec $\mathcal{A}_{intra} = \{(i,j) : |y_i - y_j| < \tau, i,j \in \{1,\dots,n\}, i \neq j\}$, $\mathcal{A}_{inter} = \{(i,j) : |y_i - y_j| \geq \tau, i,j \in \{1,\dots,n\}, i \neq j\}$, $n_{intra} = card(\mathcal{A}_{intra})$ et $n_{inter} = card(\mathcal{A}_{inter})$. La fonction f est une fonction de pondération positive qui décroit avec l'amplitude de la différence. Kwak *et al.* [2008] ont montré que cette méthode donnait de bons résultats pour l'estimation de la pose de la tête en comparaison avec des méthodes classiques de réduction de dimension telles que la *Sliced Inverse Regression* (SIR, Li 1991), ou les *Principal Hessian Directions* (PHD, Li 1992).

Plusieurs solutions ont été proposées pour intégrer l'information des labels dans les techniques d'apprentissage de la variété. SE-Isomap [Li et Guo, 2006] construit tout d'abord

une matrice des distances géodésiques pour chaque classe. Les différentes matrices sont ensuite regroupées au sein de la matrice des distances globales discriminantes. Pour finir, cette matrice est utilisée dans l'étape de réduction de dimension par MDS. Dans [Yan *et al.*, 2007], le graphe intrinsèque caractérise la compacité intra-classe et connecte les échantillons de même classe, alors que le graphe de pénalité connecte les points à la marge et caractérise la séparabilité inter-classe. La *Marginal Fisher Analysis* cherche une transformation qui maximise la compacité intra-classe et la séparabilité inter-classe.

De Ridder *et al.* [2003] utilisent une métrique qui prend en compte les labels associés aux données pour mesurer les distances entre échantillons. Cette métrique, qui augmente les distances inter-classes et diminue les distances intra-classes, remplace la distance euclidienne lors de la construction du graphe d'adjacence de la LLE. La méthode WheightedIso de Vlachos *et al.* [2002] joue également sur la métrique et multiplie par un facteur constant $\lambda < 1$ la distance entre deux échantillons d'une même classe. Geng *et al.* [2005] ont montré que la qualité des résultats était très dépendante du choix de λ. Ils définissent une métrique telle que la distance entre deux échantillons de la même classe soit toujours inférieure à 1 et que la distance entre deux échantillons de classes différentes soit toujours supérieure à $1-\alpha$. Le coefficient α, défini empiriquement, a été introduit pour permettre à deux échantillons de classes différentes de se « ressembler » plus que certains échantillons de la même classe et pour limiter les risques de déconnexion dans le graphe d'adjacence. De même que la LDA, ces méthodes concernent les problèmes de classification. Le *Biased Manifold Embedding* [Balasubramanian *et al.*, 2008] s'adresse principalement aux problèmes de régression en pondérant chaque distance inter-exemples, par un facteur qui dépend de la différence des labels. Cette méthode a donné de bons résultats pour l'estimation de la pose de la tête.

Une autre approche supervisée de réduction de dimension consiste à définir la structure idéale de la variété et d'apprendre la transformation vers cette structure topologique. Lee et Elgammal [2006] modélisent la marche d'un humain par un tore. Une dimension est utilisée pour décrire la configuration du corps et une autre dimension représente le point de vue de la caméra. La relation entre une silhouette et sa position sur le tore est apprise à l'aide d'un réseau de neurones à base radiale.

1.2.3 Synthèse

Le tableau 1.1 résume, par ordre chronologique, les différentes techniques de réduction de dimension et d'apprentissage de variétés. Il spécifie si la projection est linéaire ou non-linéaire, supervisée ou non-supervisée et s'il existe une manière de projeter sur l'espace réduit un exemple qui n'appartient pas à l'ensemble d'apprentissage (*out of sample*). La dernière colonne précise si la méthode cherche à préserver des propriétés géométriques locales ou globales lors de la projection. Les méthodes globales reflètent plus fidèlement la structure globale des données, tandis que les méthodes locales sont plus faciles à calculer (matrice creuse) et plus à même de modéliser les variétés dont la géométrie est localement proche de la géométrie euclidienne, mais dont la géométrie globale ne l'est pas [de Silva et Tenenbaum, 2002].

TABLE 1.1 – Récapitulatif des différentes techniques de réduction de dimension.

Méthodes	Lineaire (L) / Non-linéaire (N)	Supervisée (S) / Non-supervisée (N)	Projection explicite (O/N)	Local (L) / Global (G) [1]
ACP [Pearson, 1901]	L	N	**O**	G
LDA [Fisher, 1936]	L	**S**	**O**	G
MDS [Torgerson, 1952]	L	N	**O**	G
SIR [Li, 1991]	L	**S**	**O**	G
KPCA [Schölkopf *et al.*, 1998]	**N**	N	**O**	G
Isomap [Tenenbaum *et al.*, 2000]	**N**	N	N	G
LLE [Roweis et Saul, 2000]	**N**	N	N	L
LE [Belkin et Niyogi, 2003]	**N**	N	N	L
SLLE [De Ridder *et al.*, 2003]	**N**	**S**	N	L
MVU [Weinberger *et al.*, 2004]	**N**	N	N	G
LTSA [Zhang et Zha, 2005]	**N**	N	N	L
SE-Isomap [Li et Guo, 2006]	**N**	**S**	N	G
GE [Yan *et al.*, 2007]	**N**	**S**	**O**	G/L
LDA-r [Kwak *et al.*, 2008]	L	**S**	**O**	G
BME [Balasubramanian *et al.*, 2008]	**N**	**S**	N	G/L

1.3 Estimation de la pose

1.3.1 Méthodes par comparaison avec des prototypes

Cette catégorie de méthodes est certainement la plus simple et la plus intuitive. Elle consiste à comparer l'apparence du visage dans une image à un ensemble de visages d'une base de données étiquetée en pose. On attribue au visage de l'image l'orientation du visage le plus ressemblant dans la base de données (*cf.* figure 1.5). On utilise couramment l'erreur quadratique moyenne [Niyogi et Freeman, 1996; La Cascia *et al.*, 2000; Sherrah et Gong, 2001], la corrélation croisée normalisée [Beymer, 1994] ou l'information mutuelle [Goudelis *et al.*, 2008] comme mesure de ressemblance.

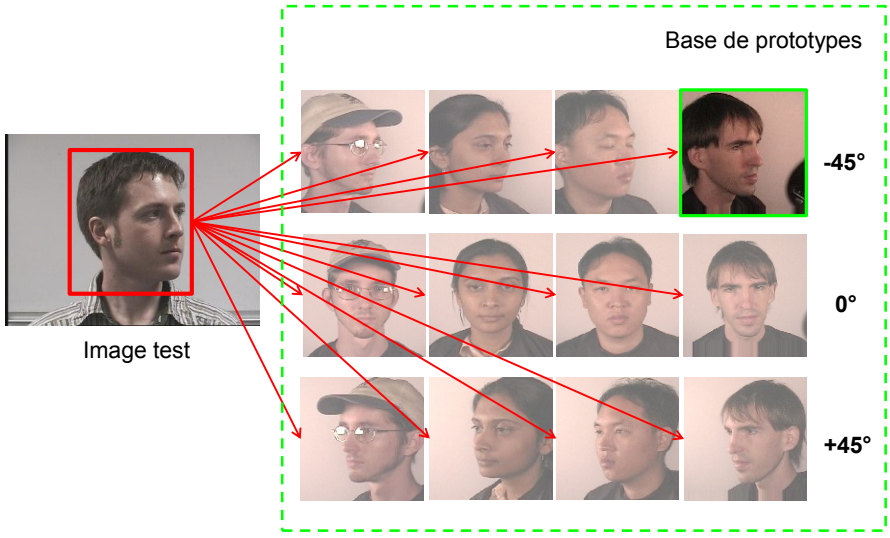


FIGURE 1.5 – Les méthodes par comparaison avec des prototypes. L'orientation d'un nouveau visage correspond à la pose du visage de la base de prototypes le plus ressemblant

Ces méthodes n'ont pas besoin d'une phase d'apprentissage et sont incrémentales en données (on peut augmenter sans effort l'ensemble des prototypes). Toutefois, des problèmes d'efficacité surviennent lorsque le nombre de prototypes est trop important puisque la complexité des ces algorithmes est linéairement dépendante du nombre d'éléments dans la base de données. Une solution à ce problème consiste à organiser les données de la base sous forme d'un arbre [Niyogi et Freeman, 1996; Sengupta *et al.*, 2002; Grujic *et al.*, 2008]. Dans [Niyogi et Freeman, 1996] par exemple, l'arbre est construit en séparant récursivement les exemples en deux groupes. A chaque nœud de l'arbre, une analyse en composante principale (ACP) détermine la direction du maximum de variance du groupe. On projette alors les exemples sur cet axe et on les sépare en deux classes en fonction du signe de leur coordonnée sur cet axe. Cette opération est répétée jusqu'à ce qu'il n'y ait plus assez d'exemples dans le groupe pour faire une ACP. Dans le cas d'un arbre binaire, le coût de la recherche diminue d'un facteur M à un facteur $log(M)$, avec M le nombre d'éléments dans

la base de données.

Dans les méthodes par comparaison avec des prototypes, le visage doit être préalablement détecté et une erreur de localisation peut fortement dégrader la précision du système. Niyogi et Freeman [1996] proposent de générer des prototypes présentant des décalages en translation et en changement d'échelle. Si l'image la plus ressemblante correpond à un prototype avec décalage, on peut facilement ajuster le cadrage de manière à avoir la ressemblance la plus forte avec un prototype centré.

Ce type d'approche s'apparente à des méthodes par suivi ; l'objectif est de maximiser la ressemblance entre l'apparence d'un modèle pour différents paramètres (translation, rotation, changement d'échelle par exemple) et l'image du visage dont on cherche à connaître la pose. Le modèle correspond le plus souvent à une image frontale du visage que l'on plaque sur un modèle géométrique simple tel qu'une ellipsoïde [Basu *et al.*, 1996] [Choi et Kim, 2008], un cylindre [La Cascia *et al.*, 2000] [Xiao *et al.*, 2003] ou un modèle générique de visage [Malciu et Prêteux, 2000]. La texture du modèle est parfois mise à jour au cours de la séquence afin de prendre en compte des changements d'illumination par exemple [Xiao *et al.*, 2003].

Par ailleurs, les méthodes par comparaison à des prototypes supposent que des visages différents dans une même pose sont toujours plus semblables qu'un même visage dans deux poses différentes. En d'autres termes, on souhaite que la mesure de similarité dans l'espace des images soit représentative de la distance dans l'espace des poses. Il est possible d'agir à deux niveaux :

- Appliquer une transformation de l'image qui accentue les caractéristiques liées à la pose et qui atténue les autres sources de variation telles que l'illumination, les expressions faciales et l'identité (*cf.* section 1.1 et 1.2).
- Choisir une mesure de ressemblance adaptée. La Cascia *et al.* introduisent par exemple un terme qui modélise l'illumination dans la fonction de coût à optimiser [La Cascia *et al.*, 2000]. Everingham et Zisserman [2005] utilisent une distance de Chamfer sur les contours orientés.

1.3.2 Méthodes par classification

L'idée principale est de regrouper par classe les visages de la base d'apprentissage qui ont des orientations proches et d'utiliser une méthode d'apprentissage pour déterminer si un visage inconnu appartient à cette classe d'orientation. Cette approche, illustrée par la figure 1.6, présente deux avantages par rapport aux méthodes par comparaison à des prototypes. D'une part, le visage n'est pas comparé à tous les exemples de la base d'apprentissage mais il est uniquement traité par les détecteurs spécifiques à chaque classe d'orientation. D'autre part, le problème de ressemblance équivalente dans l'espace image et l'espace des poses évoqué précédemment est pris en charge par l'algorithme d'apprentissage. La plupart des algorithmes proposés dans cette partie sont capables de traiter conjointement les problèmes de détection du visage et d'estimation de la pose, ce qui constitue un avantage certain sur

les autres approches.

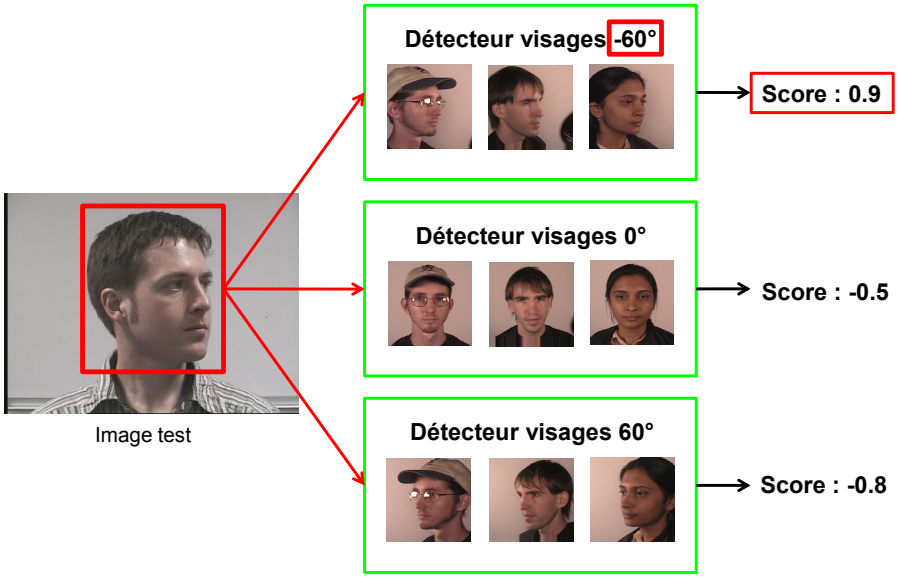


FIGURE 1.6 – Les méthodes par classification : un détecteur est appris par classe d'orientation et la pose d'un nouveau visage est donnée par le détecteur ayant obtenu le meilleur score.

Différentes techniques d'apprentissage ont été utilisées. Dans [Schiele et Waibel, 1995] et plus récemment [Voit *et al.*, 2007] chaque sortie d'un Perceptron Multicouche (MLP de l'anglais *Multi Layer Perceptron*) correspond à une classe d'orientation. La sortie est active lorsque le visage présenté en entrée correspond à cette classe. Gourier *et al.* [2007] entraînent autant de réseaux de neurones qu'il y a de classes. Chaque réseau est une Mémoire Auto-Associative Linéaire qui apprend à synthétiser en sortie, une image ressemblant à celle en entrée. Si le visage en entrée a une orientation proche de celle des visages utilisés pour l'apprentissage, les images en entrée et en sortie seront ressemblantes, sinon elles seront différentes. La pose est alors donnée par le réseau qui a obtenu l'erreur de reconstruction la plus faible. Dans [Féraud *et al.*, 2001], le procédé est semblable mais chaque classe d'orientation est modélisée par un réseau auto-associatif non linéaire et la décision finale est donnée par un MLP en fonction des sorties de chaque réseau. Sauquet *et al.* [2005] et Rowley *et al.* [1998] font également appel à des réseaux MLP. D'autres techniques d'apprentissage sont utilisées telles que la classification bayesienne [Wu et Toyama, 2000] ou les Séparateurs à Vaste Marge (SVM) [Yan *et al.*, 2001], [Li *et al.*, 2004] [Ma *et al.*, 2006a] [Guo *et al.*, 2008]. Depuis le succès du détecteur de visages frontaux de Viola et Jones [2004], de nombreux travaux ont cherché à étendre cette technique à la détection de visages multi-orientations [Jones et Viola, 2003], [Wu *et al.*, 2004], [Li et Zhang, 2004], [Huang *et al.*, 2007a].

La solution la plus directe consiste à faire fonctionner plusieurs détecteurs en parallèle (*cf.* figure 1.7(a)), chacun étant spécialisé pour une classe d'orientation [Baluja *et al.*, 2004], [Wu *et al.*, 2004]. Cette architecture présente trois inconvénients majeurs. Le premier

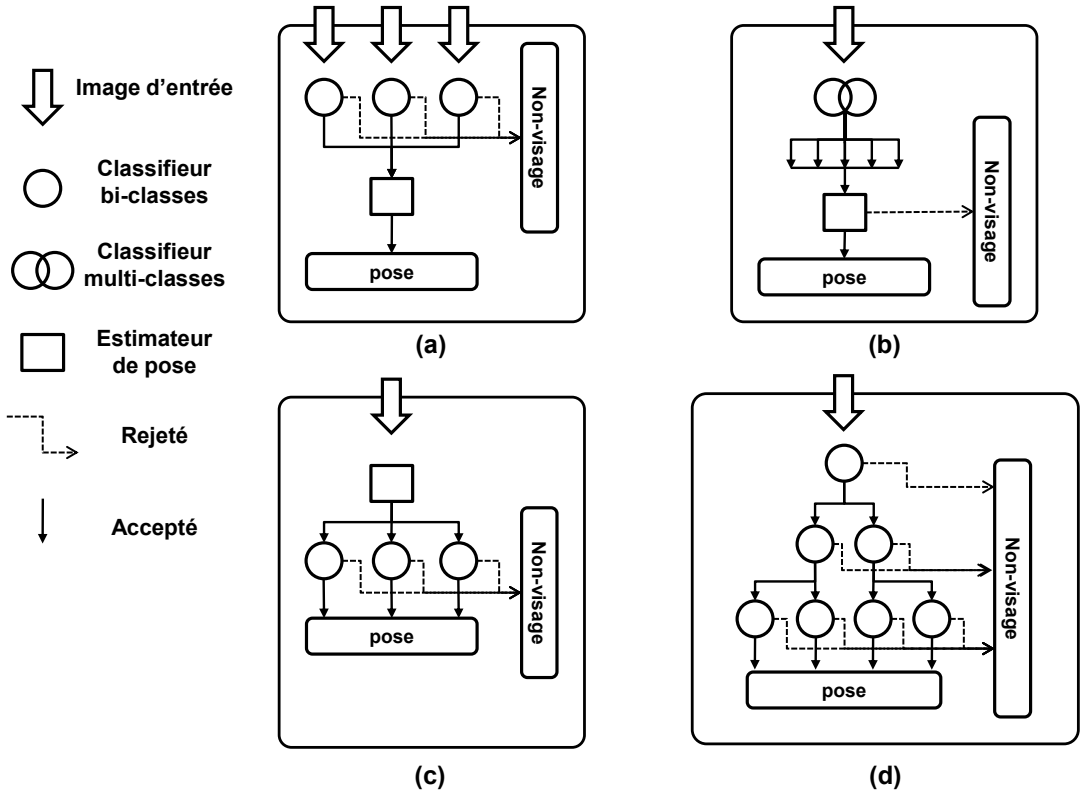


FIGURE 1.7 – Différentes architectures pour l'estimation de la pose par classification. (a) En parallèle, (b) multi-classes, (c) avec routeur, (d) descendante.

concerne la prise de décision finale puisque différents classifieurs peuvent répondre positivement. Lorsque l'on peut associer une mesure de confiance à la sortie de chaque classifieur, il est possible d'appliquer des règles simples de décision telles qu'une moyenne pondérée ou la règle du « *winner takes all* » (la pose est attribuée en fonction du classifieur ayant répondu avec le plus de certitudes) [Wu et Toyama, 2000] [Gourier *et al.*, 2007] [Wu *et al.*, 2004]. Toutefois, trouver la relation entre la sortie des classifieurs et la pose n'est pas toujours si évidente et certains ont recours à un MLP [Féraud *et al.*, 2001] ou un SVM [Yan *et al.*, 2001]. Cet expert est également utilisé pour décider si l'image correspond ou non à un visage. Le deuxième problème a trait à son efficacité puisque à chaque classification, il faut faire appel à tous les classifieurs. Le partitionnement de l'ensemble d'apprentissage constitue un autre inconvénient de cette architecture. En effet, il n'est pas évident de déterminer les frontières des différentes classes puisque les changements de pose ne sont pas discrets mais continus. De plus, le nombre d'exemples disponibles pour chaque classifieur est inversement proportionnel au nombre de poses que l'on pourra discriminer.

Dans une architecture multi-classes [Schiele et Waibel, 1995] [Zhao *et al.*, 2002] [Voit *et al.*, 2007], tous les exemples sont utilisés pour apprendre simultanément toutes les classes d'orientation (*cf.* figure 1.7(b)). Toutefois le problème de la prise de décision reste ouvert lorsque plusieurs sorties du classifieur répondent positivement. Schiele et Waibel [1995] et Zhao *et al.* [2002] utilisent une représentation gaussienne en sortie ; non seulement la sortie désirée est active, mais les sorties correspondant à des orientations proches sont également actives avec une intensité moindre. Cela permet de prendre partiellement en compte la continuité des valeurs de pose.

Une autre stratégie est d'inverser les étapes de détection de visage et d'estimation de pose [Rowley *et al.*, 1998] [Li *et al.*, 2004]. Considérant que l'image en entrée est un visage, l'algorithme estime sa pose et l'envoie au détecteur concerné qui décide si l'image correspond effectivement à un visage (*cf.* figure 1.7(c)). Ainsi, un seul détecteur est requis à chaque estimation. Rowley *et al.* [1998] estiment la rotation dans le plan et appliquent la rotation inverse à l'image afin que le visage se retrouve dans une position canonique. Le même détecteur de visages frontaux peut être utilisé pour n'importe quelle pose (cf. figure 1.8). Le principal problème est que la fiabilité de ces méthodes repose sur les performances de l'estimateur de pose en amont. De plus, l'approche de Rowley *et al.* ne peut pas s'étendre aux rotations hors-plan du visage.

Jones et Viola [2003] proposent de diviser cette tâche en une succession de décisions simples binaires structurée en arbre. Cette solution appartient aux approches descendantes (*coarse to fine* en anglais) illustrée sur la figure 1.7(d). Il existe différentes stratégies descendantes dans la littérature. Dans [Li et Zhang, 2004], par exemple, l'espace des visages est subdivisé en sous-espaces de plus en plus réduits. Dans les premiers niveaux de la pyramide, les visages d'un sous-espace présentent de grandes variations de pose mais sont traités comme un ensemble afin d'extraire leurs caractéristiques communes et de les différencier des non-visages. Dans les étages inférieurs, les caractéristiques sélectionnées sont discriminantes par rapport à la pose du visage. A chaque étage de la pyramide, les visages sont soit considérés comme des non-visages, soit transmis à tous les classifieurs du niveau

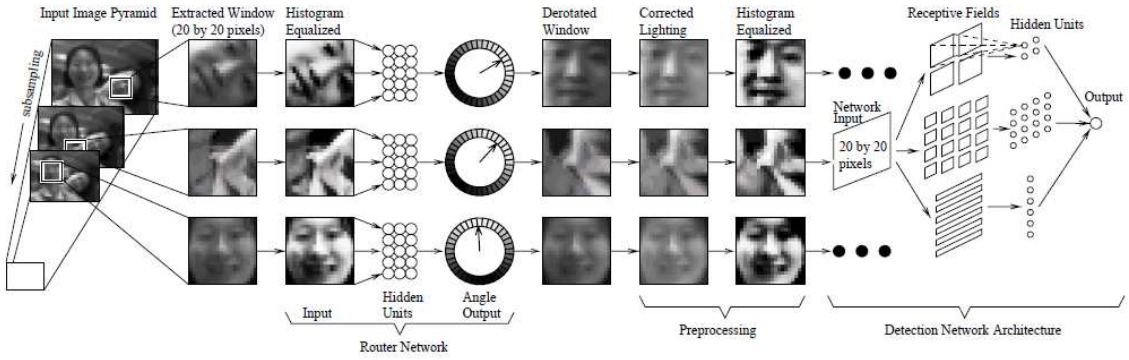


FIGURE 1.8 – Architecture par routeur proposée par Rowley *et al.* [1998].

suivant. Huang *et al.* [2007a] proposent une solution intermédiaire dans laquelle le visage à analyser est transmis à un nombre variable de détecteurs de l'étage suivant. La figure 1.7(d) présente une arborescence générique pour les stratégies « approche-précision » qui ne tient pas compte de toutes les subtilités structurelles exposées dans [Huang *et al.*, 2007a].

TABLE 1.2 – Comparaison des différentes architectures présentées dans la figure 1.7

	Parallèle	Multi-classes	Routeur	approche-précision
Partitionnement des données	− −	++	− −	+
Temps de calcul	− −	++	++	+
Simplicité de mise en oeuvre	++	++	++	− −
Précision	− −	−	− −	−
Détection de la tête	+	− −	+	++

Le tableau 1.2 synthétise les points forts et les points faibles de chaque architecture. Les méthodes par classification sont performantes car elles s'appuient sur des méthodes d'apprentissage robustes et éprouvées telles qu'AdaBoost ou les SVM. De plus, elles combinent, pour la plupart, deux tâches distinctes : la détection de visages et l'estimation de la pose. Cette dernière propriété peut toutefois être un problème car les deux tâches s'opposent ; la détection cherche à séparer les visages des non-visages en s'appuyant sur des caractéristiques partagées par tous les visages quelle que soit l'orientation alors que l'estimation de pose cherche justement à les différencier. Osadchy *et al.* [2007] ont proposé une méthode d'unification des deux tâches. Un réseau de neurones à convolution apprend à projeter un visage sur une variété paramétrée par la pose et à garder les non-visages le plus éloigné possible de cette variété.

1.3.3 Méthodes par régression

Nous avons vu dans la partie précédente que le principal problème était le partitionnement des données. L'objectif des approches par régression est d'apprendre une relation

fonctionnelle entre l'apparence d'un visage et sa pose. Ce principe est illustré par la figure 1.9. La relation est modélisée à partir d'un ensemble d'apprentissage étiqueté et fournit une estimation continue de la pose pour n'importe quel nouveau visage. Cette tâche est complexe car il s'agit d'approximer une fonction fortement non-linéaire dans un espace de grande dimension.

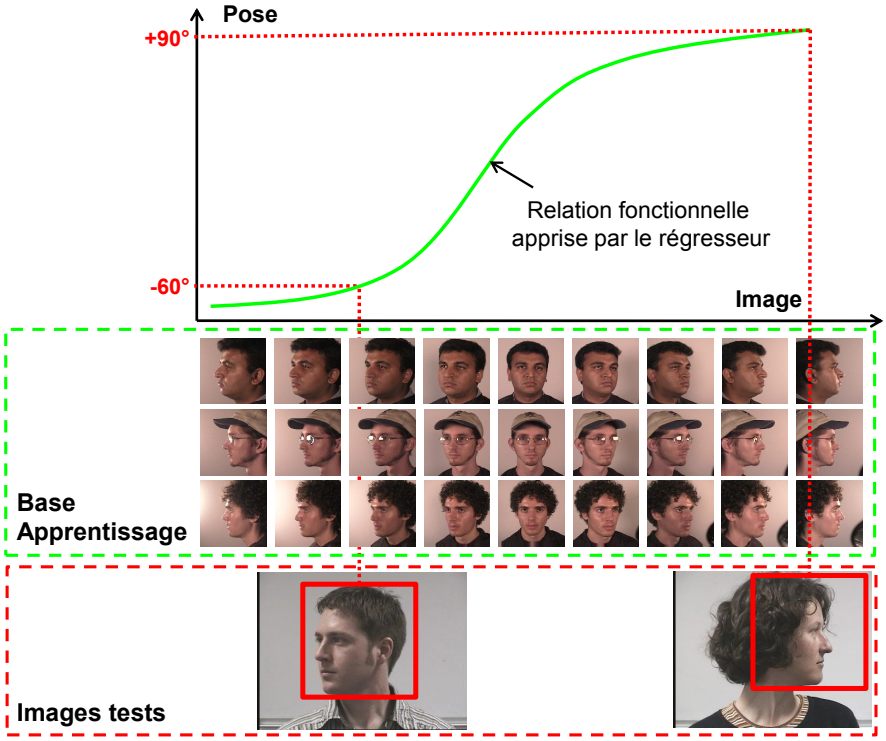


FIGURE 1.9 – Les méthodes par régression non-linéaire visent à apprendre une relation fonctionnelle entre l'apparence d'un visage et sa pose.

Li *et al.* [2004], Murphy-Chutorian *et al.* [2007] ou Guo *et al.* [2008] utilisent avec succès une *Support Vector Regression* (SVR) pour apprendre cette fonction. Li *et al.* filtrent au préalable l'image par un opérateur de Sobel et réduisent la dimension des images filtrées par une ACP. Murphy-Chutorian *et al.* utilisent la même méthode de représentation que SIFT [Lowe, 2004], une concaténation d'histogrammes des orientations de contours pour différentes zones de l'image.

La régression est le plus souvent apprise par un réseau de neurones. Voit *et al.* [2007] entraînent un MLP (*cf.* section 1.5.2) ; l'activation d'une sortie est proportionnelle à l'amplitude de la rotation sur un des degrés de liberté de la tête.

Rae et Ritter [1998] utilisent une LLM (*Locally Linear Map*) pour apprendre la relation entre l'espace image et l'espace des poses. L'idée est de subdiviser l'espace d'entrée en régions représentées par des prototypes et d'apprendre une relation linéaire entre la distance d'un exemple à ce prototype et les paramètres de pose (figure 1.10). Le processus

d'estimation de la pose d'un visage inconnu consiste à chercher le prototype le plus proche et d'appliquer la régression linéaire correspondante. Cette approche a été étendue à des exemples dont la dimension à été réduite par une ACP [Bruske *et al.*, 1998] et à des visages représentés par des ondelettes de Gabor [Krüger et Sommer, 2002].

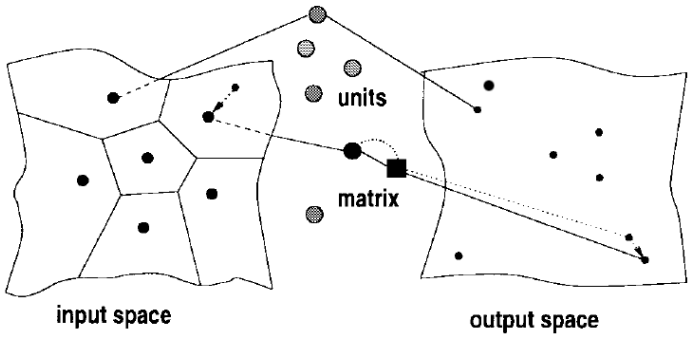


FIGURE 1.10 – Architecture d'un réseau LLM tirée de [Rae et Ritter, 1998].

Certaines méthodes n'apprennent pas la relation entre l'image et l'espace des poses mais une projection entre l'espace image et un sous-espace de faible dimension. Dans [Hu *et al.*, 2005], un réseau RBF (*Radial Basis Function*) apprend la projection sur une variété à deux dimensions trouvée par Isomap. De même, Balasubramanian *et al.* [2008] utilisent un GRNN (*Generalized Regression Neural Network*) pour apprendre la projection vers un espace réduit.

Les méthodes par régression sont rapides à calculer, n'ont besoin que d'images de visages cadrés (par opposition aux méthodes de classification qui nécessitent également des contre-exemples) et sont les méthodes globales les plus précises [Murphy-Chutorian et Trivedi, 2009]. Toutefois, ces méthodes réalisent une estimation de la pose indépendamment de l'étape de détection du visage et sont dépendantes de la qualité de ce cadrage.

1.4 Jeux de données

1.4.1 Pointing 04

Pointing 04 [Gourier *et al.*, 2004a] est une base de données publique disponible sur Internet [2]. Elle a été constituée à l'occasion du Workshop ICPR [3] Pointing 04 : *Workshop on Visual Observation of Deictic Gestures*. Elle a également été utilisée pour évaluer les méthodes d'estimation de la pose de la tête lors de l'évaluation CLEAR 07 : *International Evaluation on Classification of Events Activities and Relationships*. Elle contient 15 personnes photographiées dans 93 poses différentes. Ce processus a été répété deux fois par personne à des moments différents. Il en résulte deux séries de données. La première est

2. http ://www-prima.inrialpes.fr/Pointing04/data-face.html
3. International Conference on Pattern Recognition

réservée à l'apprentissage et le seconde aux tests. Il y a de grandes variations entre les individus : hommes/femmes, avec/sans lunettes, différentes couleurs de peau, moustaches et barbes... La pose est déterminée par les angles pan et tilt. L'angle pan varie de -90° à +90° et l'angle tilt varie de -60° à +60°. La figure 1.11 illustre la variété de la base.

FIGURE 1.11 – Echantillon d'images de la base Pointing.

L'étiquetage est obtenu en demandant aux personnes de regarder des marqueurs répartis dans la pièce. Ce mode opératoire induit une incertitude sur la vérité terrain de l'order de 15° (*cf.* figure 1.12) puisque la direction du regard ne coïncide pas toujours avec la direction de la tête.

FIGURE 1.12 – Incertitude sur l'étiquetage des données : entre chaque paire d'images, la variation supposée d'angle est de 15°.

1.4.2 FacePix

FacePix [Little *et al.*, 2005] est le second jeu de données que nous utilisons pour nos tests. Il s'agit d'une base de données publique constituée par le *Center for Cognitive Ubiquitous Computing* (CUbiC) de l'université de l'Arizona, disponible gratuitement sur Internet [4]. Elle est composée de trois ensembles de données : un ensemble contient des variations de

4. http ://www.facepix.org/

prises de vue, les deux autres présentent des variations d'illumination. Chaque ensemble est formé de 181 images (représentant des variations d'angle de -90° à +90° avec un pas de 1°) de 30 personnes différentes. Chaque image est de taille 128 × 128 pixels. Les images sont normalisées de manière à ce que tous les yeux soient alignés verticalement et que la distance en pixels entre les yeux et la bouche soit constante pour tous les individus. De plus, les visages sont centrés horizontalement afin que les yeux, le nez et la bouche restent au centre de l'image. Cette dernière contrainte est très subjective, en particulier lorsque le visage est de profil et l'on observe des différences entre individus.

Dans l'expérience proposée à la section 3.7 page 75, nous avons utilisé uniquement un ensemble de données présentant des variations de pose de -90° à +90° avec un pas de 5° (*cf.* figure 1.13). 20 visages ont été sélectionnés aléatoirement pour l'apprentissage, 6 pour les tests, et 3 pour la validation. L'acquisition est effectuée à l'aide d'une caméra en rotation autour d'un visage fixe. Le processus d'étiquetage des données est donc précis. Contrairement à Pointing 04, seul l'angle horizontal varie.

FIGURE 1.13 – Echantillon de poses de la base FacePix avec un pas de rotation de 5°.

1.5 Méthodes de comparaison

Dans la section 1.5.2, nous présentons les méthodes proposées lors de la campagne d'évaluation CLEAR 2006 que nous prendrons comme références pour situer notre approche. Nous nous comparons également à un réseau de neurones à convolution dont l'architecture est présentée à la section 1.5.1.

1.5.1 Réseau de Neurones à Convolution

Nous avons entraîné un réseau de neurones à convolution (CNN de l'anglais *Convolutional Neural Network*) proposé initialement par Le Cun *et al.* [1990]. Ce choix a été motivé principalement par deux raisons : il intègre implicitement une phase d'extraction de caractéristiques et il a été utilisé avec succès dans de nombreuses applications [Garcia et Delakis, 2004; Le Cun *et al.*, 1998; Osadchy *et al.*, 2007]. Le CNN est un réseau multicouche bioinspiré [Hubel et Wiesel, 1962] qui combine trois idées principales : les champs récepteurs locaux, les poids partagés et le sous-échantillonnage. Dans une architecture classique,

chaque cellule des couches de convolution est connectée à un ensemble de cellules regroupées dans un voisinage rectangulaire sur la couche précédente. Les champs récepteurs locaux permettent d'extraire des caractéristiques basiques tels que des contours par exemple. Les couches sont dites « à convolution » car les poids sont partagés et chaque cellule de la couche réalise la même combinaison linéaire (avant d'appliquer la fonction sigmoïde) qui peut être vue comme une simple convolution. Ces caractéristiques sont alors combinées à la couche suivante afin de détecter des caractéristiques de plus haut niveau. Entre deux phases d'extraction de caractéristiques, le réseau réduit la résolution de la carte des caractéristiques par un moyennage et un sous-échantillonnage. Cette réduction se justifie à deux titres : diminuer la taille de la couche et apporter de la robustesse par rapport aux faibles distorsions. La figure 1.14 décrit l'architecture que nous avons développée en suivant les recommandations de Simard *et al.* [2003].

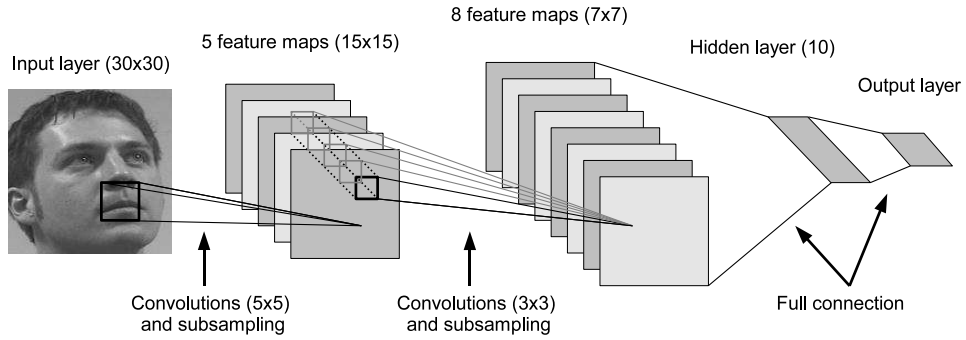


FIGURE 1.14 – Architecture de notre réseau de neurones à convolution

Le réseau est entraîné avec un algorithme classique de rétropropagation. Nous avons utilisé un pas adaptatif et une régularisation du gradient par un *momentum*.

1.5.2 Méthodes de l'évaluation CLEAR 2007

Nous présentons dans cette partie, les méthodes proposées lors de la campagne d'évaluation CLEAR'07 [Stiefelhagen et Garofolo, 2007]. Trois méthodes ont obtenu des résultats sur la base Pointing 04 et serviront de références pour comparer nos algorithmes.

1.5.2.1 Approche par un perceptron multicouches (MLP, *Multi Layer Perceptron*)

Voit *et al.* [2007] proposent une approche en deux étapes. Dans un premier temps, un classifieur détecte les pixels de teinte chair. Il s'agit d'un classifieur dont la frontière de décision est linéaire dans l'espace colorimétrique Teinte Saturation Valeur (HSV, de l'anglais *Hue Saturation Value*). La zone du visage correspond à la boîte englobante de la plus grande composante connexe. Dans un second temps, la pose de la tête est estimée par un réseau MLP. L'image du visage est redimensionnée en 64×64 pixels, convertie en niveaux de gris et l'histogramme est égalisé. On extrait une carte des modules des gradients de l'image.

L'entrée du réseau de neurones correspond à la concaténation de l'image du visage et de la carte des modules. Le réseau a une sortie par angle à estimer. La figure 1.15 illustre l'architecture du MLP.

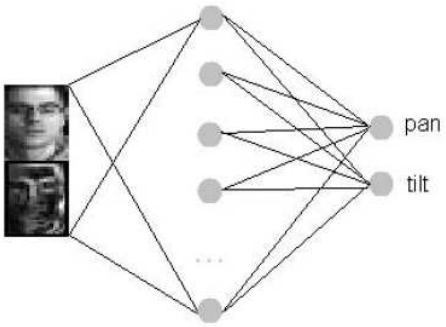


FIGURE 1.15 – Architecture du réseau MLP proposée par Voit *et al.* [2007].

1.5.2.2 Approche par des mémoires auto-associatives linéaires (MAAL)

Gourier *et al.* [2007] séparent également l'étape de détection du visage de celle d'estimation de la pose. La première étape, décrite en détail dans [Gourier *et al.*, 2004a] consiste à détecter les pixels de teinte chair par une décision bayesienne. La densité de probabilité conditionnelle d'un pixel d'appartenir à une région de peau peut être estimée en utilisant un histogramme de la chrominance. Le calcul de la chrominance (r, g) d'un pixel (x, y) est effectué en normalisant les composantes rouge et verte du vecteur de couleur (R, G, B) par son intensité lumineuse $R+G+B$. La probabilité $p((x, y) \in Peau|r, g)$ pour un pixel (x, y) de chrominance (r, g) d'être de teinte chair est donnée par :

$$p((x, y) \in Peau|r, g) = \frac{Histogramme_{peau}(r, g)}{Histogramme_{image}(r, g)} \tag{1.8}$$

Les probabilités sont calculées pour tous les pixels de l'image. Les premiers et deuxièmes moments de ces cartes de probabilités donnent la position et l'orientation et le facteur d'échelle du visage dans le plan image. Le visage est alors normalisé en taille et en inclinaison dans une vignette de 23×30 pixels en niveaux de gris (cf. figure 1.16)

FIGURE 1.16 – Détection et normalisation des visages proposées par Gourier *et al.* [2004a]

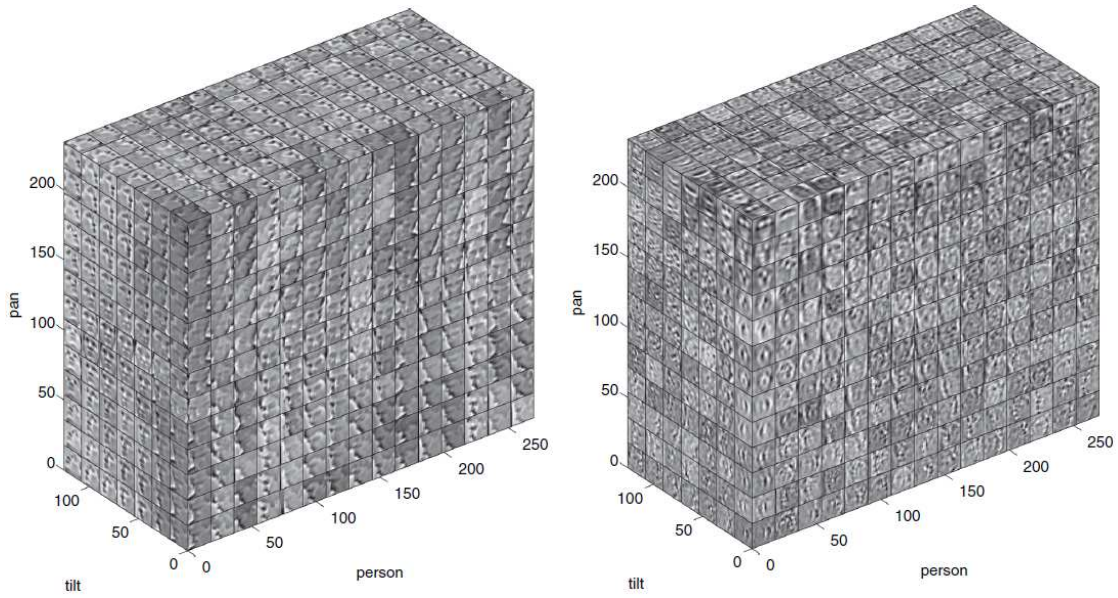
Dans une deuxième étape, une MAAL est apprise par orientation. Ainsi, lorsqu'on présente en entrée une image $\mathbf{x}$ dont la pose p correspond à celle apprise par la MAAL, l'image

reconstruite $\mathbf{x}_p$ en sortie sera très semblable à $\mathbf{x}$. Le processus d'estimation de pose consiste donc à sélectionner le réseau qui maximise la ressemblance entre l'image source et l'image reconstruite :

$$Pose = \arg\max_p(\cos(\mathbf{x}, \mathbf{x}_p)) \tag{1.9}$$

1.5.2.3 Approche par un modèle tensoriel

Cette méthode a été proposée par Tu *et al.* [2007]. Pendant la phase d'apprentissage, le bout du nez de chaque visage est localisé manuellement et une vignette de taille 18×18 pixels est extraite autour de cette position. Les vignettes de la base d'apprentissage sont organisées dans un tableau de dimension $324 \times 15 \times 13 \times 7$ correspondant respectivement au nombre de pixels, d'identités, de variations de l'angle pan, et de variation de l'angle tilt (*cf.* figure 1.17(a)). Le modèle tensoriel, illustré sur la figure 1.17(b) est créé à l'aide d'une analyse en composantes indépendantes multilinéaire, comme proposé par Vasilescu et Terzopoulos [2005].



(a) (b)

FIGURE 1.17 – Construction du modèle tensoriel. (a) Organisation matricielle des vignettes. (b) Modèle tensoriel proposé par Vasilescu et Terzopoulos [2005]

En test, le visage est grossièrement localisé à l'aide du modèle de teinte chair proposé par Jones et Rehg [2002] et la pose est inférée en projetant l'image sur le modèle tensoriel (*cf.* figure 1.18). La localisation est affinée en maximisant la corrélation entre l'image extraite et l'image reconstruite à l'aide du modèle tensoriel. Pour finir, la pose est affinée par un algorithme du plus proche voisin dans un espace réduit par une analyse en composante

principale (ACP).

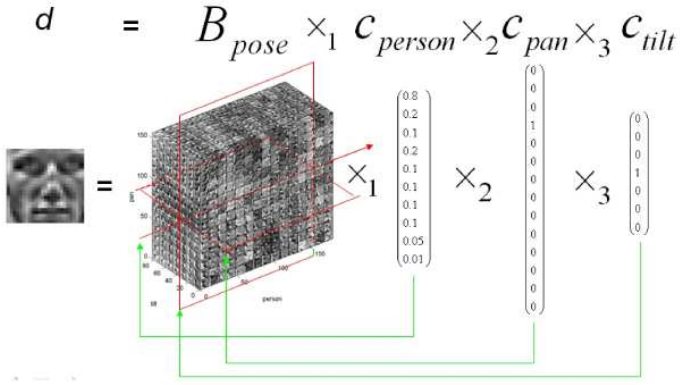


FIGURE 1.18 – Illustration de l'estimation de la pose par projection sur le modèle tensoriel tirée de Tu *et al.* [2007]

Il existe d'autres méthodes qui ont été testées sur la base pointing 04 [Li *et al.*, 2007b], [Stiefelhagen, 2004] mais nous ne les prenons pas en compte dans nos comparaisons car elles ne suivent pas le protocole défini pour l'évaluation CLEAR. Dans [Stiefelhagen, 2004] par exemple, 80% des exemples sont utilisés en apprentisssage, 10% en validation et seulement 10% en test. Li *et al.* [2007b] utilisent toute l'image et ne pré-sélectionnent pas la zone du visage. Par ailleurs leurs résultats sont moins bons quelle que soit la méthode de réduction de dimension envisagée.

1.5.3 Conclusion

Dans ce chapitre consacré à l'état de l'art, nous avons décrit la chaîne de traitement des méthodes globales d'estimation de la pose de la tête. Nous avons focalisé notre analyse sur l'étape de représentation des images et l'étape de décision que nous avons divisée en trois catégories : les approches par comparaison avec des prototypes, par classification et par régression.

L'état de l'art a révélé la simplicité de mise en œuvre et le caractère intuitif des méthodes par comparaison avec des prototypes. Nous proposerons dans le chapitre suivant, la mise en œvre d'une telle technique.

Nous nous sommes également intéressé aux méthodes par régression car elles sont à la fois rapides et plus précises que les autres approches globales. Le chapitre 3 décrira notre méthode et montrera son efficacité sur les jeux de données que nous venons de présenter.

Chapitre 2

Estimation de la pose par comparaison avec des images de synthèse

2.1 Approche proposée

La plupart des méthodes présentées dans l'état de l'art nécessitent une phase d'apprentissage gourmande en temps de calcul et en nombre de données. Nous nous sommes donc tournés vers les méthodes par comparaison avec des prototypes qui nous dispensent de la phase d'apprentissage. Cette approche nécessite toutefois la constitution d'une large base de données de visages présentant de grandes variations de poses, d'identités, d'illuminations et d'expressions faciales [Grujic *et al.*, 2008]. Nous contournons ce problème en générant une base de données d'exemples synthétiques. A partir d'une image frontale de la personne et d'un ensemble de points caractéristiques identifiés sur le visage, nous estimons les paramètres morphologiques et d'expression d'un modèle 3D et nous transférons la texture du visage sur ce modèle. Dans cette première étape hors-ligne, nous pouvons ainsi générer un ensemble de vues synthétiques de ce modèle. Nous estimons dans une seconde étape en ligne, la pose du visage dans une image en le comparant aux images de synthèse précédemment générées. Cette mesure de ressemblance s'appuie sur une mesure de distance entre contours orientés. Cette méthode [Bailly et Milgram, 2008a] est illustrée en figure 2.1.

2.2 Constitution de la base de données

La constitution de la base de données d'images de synthèse nécessite la construction d'un modèle 3D du visage. Nous proposons une méthode d'estimation des paramètres de forme et de pose d'un modèle déformable à partir d'un ensemble de points 2D caractéristiques identifiés dans une image monoculaire [Bailly et Milgram, 2008b]. De nombreuses solutions ont été proposées et sont référencées dans la littérature sous le terme anglais de *Structure from Motion*, Beardsley *et al.* [1997]; Cornou *et al.* [2002]). Tomasi et Kanade [1992] ont développé une méthode par factorisation pour estimer les paramètres de forme et de mouvements d'objets rigides dans des séquences d'images. Une Décomposition en Valeurs Singulières (ou SVD de l'anglais *Singular Value Decomposition*) est appliquée à la

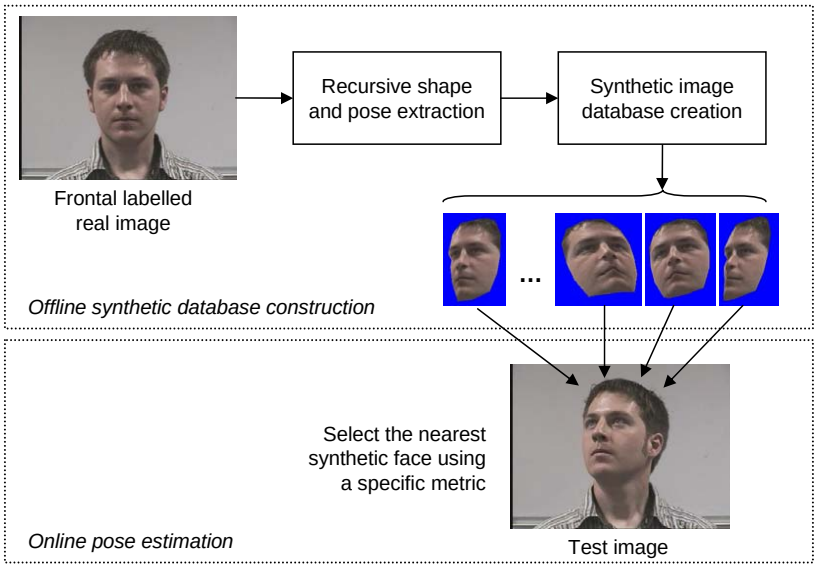


FIGURE 2.1 – Principe général de l'estimation de la pose par comparaison avec des images de synthèse

matrice qui regroupe l'ensemble des positions des points caractéristiques appariés dans un ensemble d'images pour séparer les paramètres de forme des paramètres de mouvement. Bregler *et al.* [2000] ont étendu cette méthode aux modèles déformables. Des recherches plus récentes utilisent l'algorithme des ajustements de faisceaux [Triggs *et al.*, 2000] pour minimiser l'erreur de reprojection. L'optimisation s'appuie principalement sur la méthode de Levenberg Marquart [Del Bue *et al.*, 2007; Aanaes et Kahl, 2002], mais d'autres méthodes telles que les algorithmes génétiques, sont également utilisées [Koo et Lam, 2008].

Nous nous intéressons plus spécifiquement aux cas où seule une image monoculaire est disponible. Chaumont et Beaumesnil [2005] proposent un algorithme en deux passes. Dans un premier temps, ils calculent une forme et une pose approximative en partant de l'hypothèse que tous les points du visage sont dans un même plan. L'estimation des paramètres du modèle est affinée dans un second temps en relâchant la contrainte de coplanarité sur les points. On passe alors d'une projection affine à une projection perspective.

La méthode que nous proposons [Bailly et Milgram, 2008b] estime séparément les paramètres de pose et de forme du modèle. Ainsi la détermination des paramètres a une solution analytique. Ces paramètres sont mis à jour itérativement, à la manière d'un algorithme d'*Expectation-Maximisation* (EM).

2.2.1 Modèles paramétrés

Description du modèle : l'approche proposée s'applique à tous les modèles 3D possédant une composante rigide et modulés par des vecteurs de déformation. Dans la littérature

de nombreux modèles statistiques [Cootes *et al.*, 1995] et géométriques [Ahlberg, 2001a] répondent à cette contrainte. Soit $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N)^T$ un modèle filaire constitué de N points. $(x_i, y_i, z_i, 1)^T$ correspond aux coordonnées homogènes du $i^{\text{ème}}$ points du modèle. Chaque occurrence de ce modèle s'exprime comme la combinaison d'un vecteur moyen [1] $\overline{\mathbf{x}}$ et la somme des M vecteurs de déformation $\mathbf{v}_j$ pondérés par les paramètres $\boldsymbol{\sigma}_j$:

$$\mathbf{x} = \overline{\mathbf{x}} + \sum_{j=1}^{M} \boldsymbol{\sigma}_j \mathbf{v}_j \tag{2.1}$$

Matriciellement, ce modèle s'écrit :

$$\mathbf{x} = \overline{\mathbf{x}} + \mathbf{S}\boldsymbol{\sigma} \tag{2.2}$$

Avec $\mathbf{S}$ une matrice $4N \times M$ contenant en colonne les vecteurs de déformation et $\boldsymbol{\sigma}$ un vecteur contenant les paramètres de forme.

Le modèle Candide-3 : dans notre étude, nous avons utilisé le modèle Candide (figure 2.2) proposé par Ahlberg [2001a]. Ce modèle, couramment utilisé pour l'analyse et la synthèse faciale [Chen et Davoine, 2006; Dornaika et Davoine, 2008; Weissenfeld *et al.*, 2006], est disponible publiquement sur Internet [Ahlberg]. Le maillage du modèle est constitué de 113 sommets. Le modèle peut être déformé suivant 24 modes de déformation. 13 modes définissent les caractéristiques morphologiques du visage (unités de forme) telles que la distance inter-oculaire, la largeur de la mâchoire ou la hauteur du front. 11 autres paramètres régissent les expressions faciales (unités d'action) telles que l'ouverture de la bouche, le haussement des sourcils ou le sourire. La figure 2.2 illustre l'effet de certains de ces modes de déformation.

2.2.2 Estimation récursive de la pose et de la forme du modèle 3D

L'objectif de la méthode est de minimiser la somme des résidus de reprojection. Il s'agit de la distance entre les points 2D identifiés $\mathbf{u'}$ dans l'image et les points projetés $\mathbf{u}$ du modèle 3D (*cf.* figure 2.3). Dans notre expérience, les points caractéristiques 2D ont été sélectionnés manuellement mais ils peuvent être localisés automatiquement dans une image frontale à l'aide d'un modèle actif de forme (ou ASM de l'anglais *Active Shape Model*) par exemple [Cootes *et al.*, 1995].

Cette minimisation nécessite d'estimer conjointement les paramètres de forme et de pose du modèle. Dans un premier temps, nous estimons grossièrement les paramètres de pose par l'algorithme POSIT [2] [Dementhon et Davis, 1995] en ne considérant que la partie rigide (le

1. Le vecteur moyen constitue la partie rigide du modèle. Dans le cas d'un modèle statistique, il s'agit de la position moyenne des points du modèle pour l'ensemble des visages ayant servi à construire le modèle.

2. POSIT (*Pose from Orthography and Scaling with ITeration*) est un algorithme d'estimation des paramètres de rotation et de translation d'un objet par rapport au repère camèra. Ces paramètres sont estimés à partir d'un ensemble de correspondances entre les points 3D de l'objet et leur projection sur le plan image.

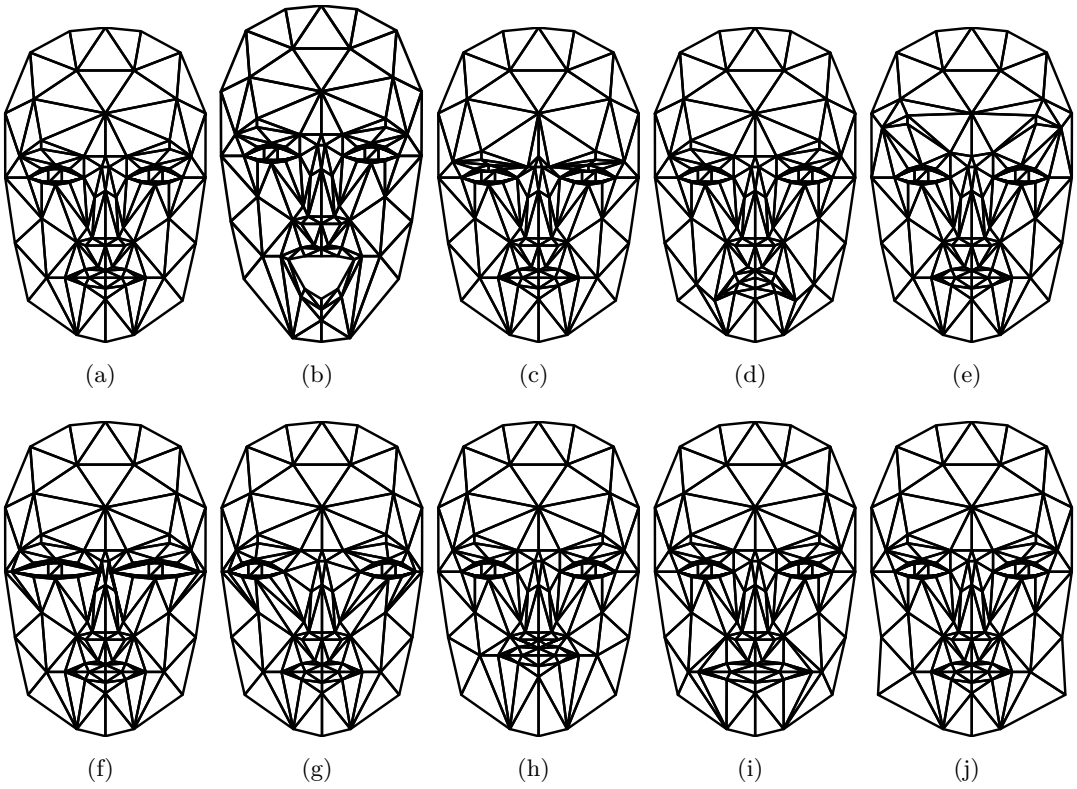


FIGURE 2.2 – Le modèle Candide-3 : La première ligne illustre des variations d'expressions et la deuxième ligne montre des variations morphologiques.

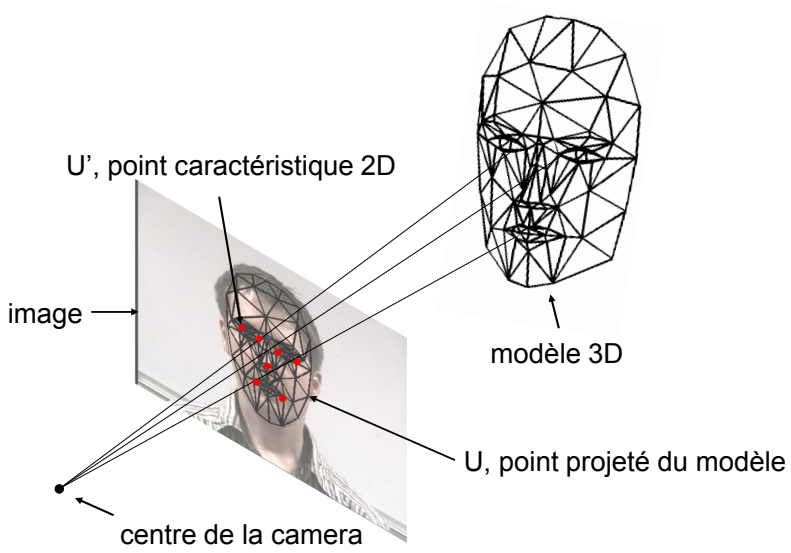


FIGURE 2.3 – Principe général : trouver les paramètres de forme et de pose qui minimisent l'erreur de reprojection.

modèle moyen $\bar{\mathbf{x}}$). Ainsi nous pouvons déterminer analytiquement les paramètres de forme dans un second temps. La forme et la pose sont alors ré-estimées itérativement tant que l'erreur de reprojection diminue.

1. **Entrées** :
   - Points caractéristiques 2D $\mathbf{u'}$
   - Modèle 3D déformable $\mathbf{x}(\boldsymbol{\sigma})$
2. **Sortie** :
   - paramètres de forme $\boldsymbol{\sigma}$
   - paramètres de pose $\mathbf{R}, \mathbf{t}$
3. **Initialisation** :
   - $\sigma \leftarrow \mathbf{0}$
4. **Tant que** l'erreur de reprojection $e < \varepsilon$ (*cf.* eq. 2.4)
   - estimation de la pose : $(\mathbf{R}, \mathbf{t}) \leftarrow$ POSIT($\mathbf{u'}$,$\boldsymbol{\sigma}$)
   - estimation analytique de la forme : $\boldsymbol{\sigma} \leftarrow$ calculeForme($\mathbf{u'}$,$\mathbf{R}$,$\mathbf{t}$)
5. **retourne** $\boldsymbol{\sigma}$, $\mathbf{R}$ et $\mathbf{t}$.

FIGURE 2.4 – Algorithme d'estimation itérative de la forme et de la pose.

2.2.2.1 Estimation de la pose du modèle

Nous ne disposons pas directement des positions 3D des points du modèle, mais uniquement de leur projection dans l'image. Dans le cas d'une transformation perspective, les coordonnées $\mathbf{u}_i = (u_i, v_i, 1)^T$ d'un point projeté du modèle s'écrivent :

$$\begin{pmatrix} su_i \\ sv_i \\ s \end{pmatrix} = \mathbf{K}\,[\mathbf{R}|\mathbf{t}]\,\mathbf{x}_i \tag{2.3}$$

Avec $\mathbf{K}$ la matrice des paramètres intrinsèques, $\mathbf{R}$ la matrice de rotation 3×3 et $\mathbf{t}$ le vecteur de translation du visage par rapport au repère de la caméra. Les paramètres intrinsèques sont supposés connus. Dans le cas contraire, ces paramètres peuvent être estimés grossièrement en fixant le point principal au milieu de l'image, en négligeant la distorsion de l'image et en fixant arbitrairement la distance focale sans grande erreur de reconstruction [Cheong et Peh, 2004].

$\mathbf{R}$ et $\mathbf{t}$ sont estimés par l'algorithme POSIT. Cet algorithme nécessite de connaître la forme du modèle. Ce dernier est initialisé par sa valeur moyenne puis l'estimation des paramètres de pose est affinée après chaque nouvelle estimation des paramètres du modèle.

2.2.2.2 Estimation des paramètres de forme du modèle

L'erreur de reprojection e entre les points images $\mathbf{u'}$ et les points projetés $\mathbf{u}$ est donnée par :

$$e = \|\mathbf{u'} - \mathbf{u}\|^2 \tag{2.4}$$

En posant $\mathbf{P} = \mathbf{K}\,[\mathbf{R}|\mathbf{t}]$, la matrice de projection perspective qui transforme un point 3D $\mathbf{x}_i$ en un point image $\mathbf{u}_i$ et en exprimant $\mathbf{x}$ (equation (2.1)), $\boldsymbol{\sigma}$ est obtenu en minimisant cette erreur :

$$\boldsymbol{\sigma}^* = \arg\min_{\boldsymbol{\sigma}} \sum_{i=1}^{N} \left\| \mathbf{u'}_i - \mathbf{P}(\overline{\mathbf{x}} + \sum_{j=1}^{M} \boldsymbol{\sigma}_j \mathbf{v}_{ij}) \right\|^2 \tag{2.5}$$

avec $\mathbf{v}_{ij}$ le $j^{\text{ème}}$ vecteur de déformation du $i^{\text{ème}}$ point du modèle.

En posant $\mathbf{Q} = \begin{pmatrix} \mathbf{P} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{P} \end{pmatrix}$ l'équation précédente peut s'écrire matriciellement de la manière suivante :

$$\boldsymbol{\sigma}^* = \arg\min_{\boldsymbol{\sigma}} \left\| \mathbf{u'} - \mathbf{Q}\,(\overline{\mathbf{x}} + \mathbf{S}\boldsymbol{\sigma}) \right\|^2 \tag{2.6}$$

Ce qui équivaut à trouver $\boldsymbol{\sigma}$ tel que

$$\mathbf{u'} = \mathbf{Q}\,(\overline{\mathbf{x}} + \mathbf{S}\boldsymbol{\sigma}) \tag{2.7}$$

La solution de cette équation s'écrit :

$$\boldsymbol{\sigma} = (\mathbf{QS})^{\dagger} \left(\mathbf{u'} - \mathbf{Q}\overline{\mathbf{x}}\right) \tag{2.8}$$

La pseudo inverse $(\mathbf{QS})^{\dagger}$ est facilement calculée par une décomposition en valeur singulière [Hartley et Zisserman, 2003]. Il est également possible de rendre le calcul plus robuste en modulant l'impact de chaque point dans l'estimation des paramètres.

$$\boldsymbol{\sigma} = \left((\mathbf{QS})^{T}\mathbf{W}(\mathbf{QS})\right)^{-1} (\mathbf{QS})^{T}\mathbf{W}(\mathbf{u'} - \mathbf{Q}\overline{\mathbf{x}}) \tag{2.9}$$

La matrice des poids $\mathbf{W}$ est estimée *a priori* en fonction de la confiance accordée à chaque point. Elle peut aussi être estimée itérativement en fonction de l'erreur résiduelle comme pour les moindres carrés repondérés itérativement [Lepetit et Fua, 2005].

Par ailleurs, la plupart des modèles ont des paramètres bornés afin de restreindre l'ensemble des formes admissibles. Les paramètres qui dépassent ces limites sont projetés sur l'espace admissible.

2.2.3 Conclusion

A l'issue de cette étape, nous avons adapté un modèle générique aux spécificités morphologiques d'un individu. En utilisant l'image frontale du visage comme texture, nous pouvons alors générer un ensemble de vues synthétiques de cette personne avec différentes poses, expressions et illuminations. Le problème d'estimation de pose se ramène donc à une mesure de ressemblance entre l'image du visage et les différentes images de la base de données synthétiques.

2.3 Comparaison des images

2.3.1 Extraction des caractéristiques

Notre mesure de ressemblance s'appuie sur l'orientation des contours. Ils présentent l'avantage d'être moins sensibles aux changements d'illumination que les niveaux de gris. Les contours orientés ont souvent été utilisés avec succès en analyse faciale [Cootes et Taylor, 2001; Le Gallou, 2007; Fröba et Küblbeck, 2002] ainsi que dans d'autres domaines tels que la détection du type de véhicule [Petrovic et Cootes, 2004; Negri *et al.*, 2006] ou l'appariement d'images [Lowe, 2004]. Les gradients horizontaux $\mathbf{G}_h$ et verticaux $\mathbf{G}_v$ de l'image $\mathbf{I}$ en niveaux de gris sont extraits par convolution avec un filtre de Sobel de taille 3×3 :

$$\mathbf{G}_h = \begin{pmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{pmatrix} * \mathbf{I} \quad \text{et} \quad \mathbf{G}_v = \begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * \mathbf{I} \tag{2.10}$$

En chaque pixel (x, y) de l'image, le module du gradient peut être approximé par :

$$\mathbf{G}(x,y) = \sqrt{\mathbf{G}_h(x,y)^2 + \mathbf{G}_v(x,y)^2} \tag{2.11}$$

Pour ne conserver que les contours, on ne considèrera qu'un pourcentage des pixels de plus fort module (typiquement, seuls 20% des pixels sont conservés).

On calcule ensuite l'orientation de chaque pixel de contour par :

$$\theta = tan^{-1}\left(\frac{\mathbf{G}_v(x,y)}{\mathbf{G}_h(x,y)}\right) \tag{2.12}$$

L'angle (modulo 2π) θ n'est pas pertinent pour comparer deux pixels de contour. En effet, deux points proches sur le cercle trigonométrique peuvent avoir des angles associés très différents (ε et $2\pi - \varepsilon$ par exemple). On adoptera donc l'approche couramment utilisée qui consiste à représenter l'angle par son cosinus et son sinus.

Par ailleurs, le fond de l'image est inconnu et peut être plus foncé ou plus clair que le visage. On souhaite pourtant que la signature qui caractérise le contour externe du visage soit constante. Il suffit de rendre cette signature π-périodique en doublant la valeur de l'angle comme dans [Cootes et Taylor, 2001].

Chaque pixel de contour sera alors représenté par un vecteur d'attributs défini par :

$$\mathbf{f}^i = \begin{pmatrix} cos(2\theta_i) \\ sin(2\theta_i) \\ x_i \\ y_i \end{pmatrix} \tag{2.13}$$

Les valeurs x_i et y_i sont normalisées dans l'intervalle $[-1\ +1]$.

2.3.2 Mesure de ressemblance

Une comparaison pixel à pixel n'est pas envisageable dans le cas d'objets déformables tels que les visages. Notre mesure de ressemblance s'inspire de la distance de Hausdorff [Rogers, 1999]. Pour chaque pixel de contour de l'image de synthèse, on calcule la distance entre son vecteur d'attributs $\mathbf{f}_s^j$ et chaque vecteur d'attributs $\mathbf{f}_r^i$ sélectionné dans l'image réelle. La distance entre deux vecteurs d'attributs est définie par :

$$d(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{\mathbf{f}_1^T \mathbf{W} \mathbf{f}_2} \tag{2.14}$$

Elle correspond à une distance euclidienne pondérée par la matrice $\mathbf{W} = diag(a, a, 1, 1)$ [3]. Le paramètre a est utilisé pour moduler l'impact de l'orientation du contour par rapport à

3. $diag(a_1, a_2, \ldots, a_n)$ est une matrice diagonale de taille $n \times n$. Les scalaires $a_1, a_2, \ldots, a_n$ correspondant aux éléments de la diagonale

sa position. Cette distance est d'autant plus petite qu'il existe dans l'image réelle un pixel de contour avec une orientation et une position semblable. Certains pixels de contour dans l'image réelle peuvent être dus au bruit. On s'intéresse alors à la médiane des distances des k pixels les plus proches. La mesure de ressemblance entre une image de synthèse $\mathbf{I}_s$ et une image réelle $\mathbf{I}_r$ est la somme de ces distances pour tous les points de contour de l'image de synthèse. Elle s'exprime par :

$$D(\mathbf{I}_s, \mathbf{I}_r) = \frac{1}{M} \sum_{j=1}^{M} \underset{i \in \mathcal{E}(j)}{mediane} \left(d(\mathbf{f}_s^j, \mathbf{f}_r^i) \right) \tag{2.15}$$

$\mathcal{E}(j)$ est l'ensemble des k plus proches voisins.

Nous ferons référence à cette métrique dans le reste du document sous le terme de Mesure Elastique sur les Contours Orientés (MECO)

2.4 Résultats

2.4.1 Protocole expérimental

Nous présentons les résultats obtenus avec des images extraites de la base de données Pointing 04. Le protocole expérimental consiste à prendre une image frontale réelle d'une personne (IFR) et d'en extraire le modèle de synthèse. On considère ensuite une image réelle de test (IRT) de la même personne dans une pose différente. On estime cette pose et on la combine avec le modèle et la texture extraite à partir de l'IFR pour produire l'image synthétique de vérité terrain (ISVT). On souhaite que l'IRT ressemble fortement à l'ISVT et faiblement aux autres occurrences du modèle de synthèse. On veut donc que $D(ISVT, IRT) < D(IS, IRT)$. IS correspond aux images synthétiques dont l'orientation diffère de la vérité terrain d'au moins 15°. Dans notre expérience nous avons généré 25 images autour de la vérité terrain avec un pas angulaire de 15° dans les directions pan et tilt. On calcule la ressemblance de chaque image de synthèse à l'image réelle de test et on ordonne les images par ordre décroissant. Le rang de l'ISVT nous renseigne sur la pertinence de notre mesure de ressemblance. Le rang 1 signifie que l'ISVT ressemble le plus à l'image réelle de test parmi toutes les images de synthèse.

2.4.1.1 Autres mesures de ressemblance

Nous avons comparé notre approche avec deux autres méthodes. La première [Petrovic et Cootes, 2004] s'appuie également sur une représentation de l'orientation des gradients mais la distance entre les points est calculée pixel à pixel. La seconde méthode [Tan et Triggs, 2007] calcule une distance qui s'inspire de la distance de Hausdorff sur des Motifs Locaux Binaires (LBP de l'anglais *Local Binary Pattern*). La mesure de ressemblance est

donc proche de celle que nous proposons mais les caractéristiques extraites de l'image sont différentes.

2.4.1.2 Méthode des *square mapped gradient* (SMG) [Petrovic et Cootes, 2004]

Chaque pixel de l'image est représenté par le cosinus et le sinus de l'orientation du gradient. Tous les pixels sont pris en compte mais ils sont modulés par une fonction de normalisation f :

$$(\mathbf{G'}_h(x,y), \mathbf{G'}_v(x,y)) = f(\mathbf{G}_h(x,y), \mathbf{G}_v(x,y))(cos(2\theta), sin(2\theta)) \tag{2.16}$$

avec

$$f(\mathbf{G}_h, \mathbf{G}_v) = \frac{\mathbf{G}}{\mathbf{G} + \overline{\mathbf{G}}} \tag{2.17}$$

$\overline{\mathbf{G}}$ est l'intensité moyenne des gradients de l'image. Les SMG sont calculés pour tous les pixels du visage dans l'image de synthèse et concaténés pour former le vecteur de caractéristiques. La mesure de ressemblance est une distance euclidienne entre ce vecteur de caractéristiques et celui extrait de l'image réelle à tester. Le nombre de pixels de visage pouvant varier, la distance est normalisée par ce nombre de pixels.

2.4.1.3 Motifs Locaux Binaires (LBP de l'anglais *Local Binary Patterns*)

Les LBP [Ojala *et al.*, 1996] sont des descripteurs locaux de textures souvent utilisés en analyse de visages [Rodriguez et Marcel, 2006; Zhao et Pietikäinen, 2007]. Ils consistent à prendre le voisinage local de chaque pixel et de le seuiller par la valeur du pixel central. Le résultat du seuillage donne une combinaison de valeurs binaires. Dans le cas d'un voisinage 3×3, on obtient un code à 8 bits.

Ojala *et al.* [2002] s'intéressent aux motifs qui n'ont, au plus, qu'une transition 0-1 et 1-0. Ces motifs sont dits « uniformes ». L'exemple de la figure 2.5 n'est pas uniforme puisque qu'il y a deux transitions 0-1 et deux transitions 1-0. Parmi les 256 LBP possibles, seuls 58 sont uniformes mais ils représentent environ 90% des patterns de l'image.

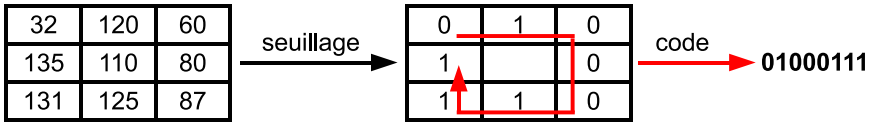


FIGURE 2.5 – Illustration du descripteur LBP.

Les LBP peuvent être instables dans les régions homogènes de l'image, c'est-à-dire lorsque les variations d'intensité par rapport au pixel central sont faibles. Tan et Triggs

[2007] proposent les motifs locaux ternaire (LTP, de l'anglais *Local Ternary Patterns*), un code à trois valeurs défini par :

$$s'(u, i_c, t) = \begin{cases} 1, & u \geq i_c + t \\ 0, & |u - i_c| < t \\ -1, & u \leq i_c - t \end{cases} \tag{2.18}$$

i_c est la valeur du pixel central. Les pixels du voisinage qui ont une intensité u semblable à $i_c \pm t$, reçoivent le code 0. Le code 1 est attribué au pixels dont l'amplitude est supérieure à $i_c + t$ et -1 pour une amplitude inférieure à $i_c - t$ (*cf.* 2.6).

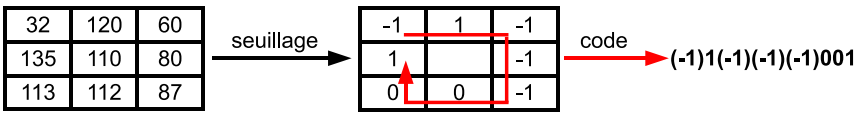


FIGURE 2.6 – Illustration du descripteur LTP pour un seuil $t = 5$.

Afin de conserver un codage binaire similaire aux LBP, le LTP est séparé en deux codes, l'un pour les valeurs positives et l'autre pour les valeurs négatives. Ainsi le code $(-1)1(-1)(-1)(-1)001$ donnera 01000001 et 10111001.

La mesure de similarité associée aux LBP et LTP s'inspire de la distance de Hausdorff. La distance entre deux images $\mathbf{I}_s$ et $\mathbf{I}_r$ est définie par :

$$D(\mathbf{I}_s, \mathbf{I}_r) = \sum_{\text{pixels } p_i \text{ de } \mathbf{I}_s} w(d^k(p_i, p_j)) \tag{2.19}$$

Pour chaque pixel p_i de $\mathbf{I}_s$, on calcule une fonction w de la distance euclidienne d^k par rapport au pixel p_j de $\mathbf{I}_r$ le plus proche dont le code k est le même que p_i. w est une fonction linéaire à seuil $w(x) = min(x, \tau)$. Le seuil τ est égal à 6 pixels dans notre expérience.

2.4.2 Résultats

Les visages de la figure 2.7 sont les images frontales réelles (IFR) utilisées pour extraire le modèle.

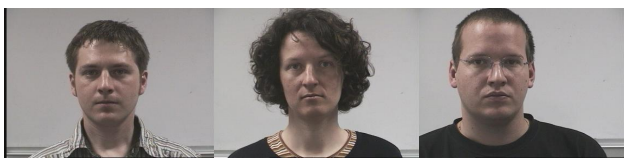


FIGURE 2.7 – Images frontales réelles (IFR)

Les images de la figure 2.8 correspondent aux images réelles de test (IRT) utilisées pour évaluer les différentes mesures de ressemblance.

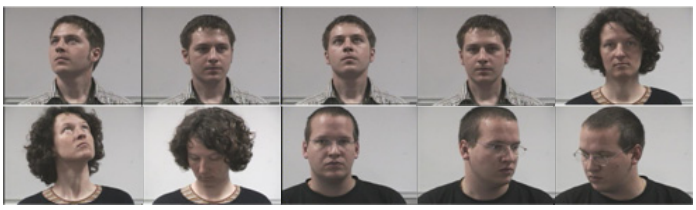


FIGURE 2.8 – Images réelles de test avec différentes poses et différentes illuminations.

Le tableau 2.1 regroupe les résultats obtenus. Le numéro correspond au rang de l'image de synthèse de vérité terrain (ISVT) parmi toutes les images de synthèse (IS) générées. Un rang élevé signifie qu'il y a des images de synthèse avec la mauvaise orientation qui ressemble plus à l'image de test que l'image de synthèse avec la bonne orientation.

# IRT	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
MECO	**2**	**3**	**1**	**2**	**1**	**2**	3	**1**	**3**	**1**
SMG	**2**	**3**	2	5	**1**	**2**	**2**	2	4	4
LBP	3	5	2	5	3	5	4	2	4	5
LTP	3	5	2	5	3	5	4	1	4	5

TABLE 2.1 – Rang de l'image de synthèse de vérité terrain. La mesure de ressemblance est d'autant plus pertinente que le rang est petit.

Notre mesure de ressemblance (MECO) obtient les meilleurs résultats pour l'ensemble des visages considérés à l'exception de la 7$^{\text{ème}}$ image (l'image est classée 3$^{\text{ème}}$ avec MECO et 2$^{\text{ème}}$ avec SMG). Le rang moyen est de 1.9 avec notre mesure contre 2.7, 3.7 et 3.8 avec SMG, LTP et LBP respectivement. On peut également remarquer que pour 40% des images, notre mesure donne la bonne réponse et que 100% des ISVT ont un rang inférieur ou égal à 3. SMG, la deuxième meilleure mesure obtient seulement 10% et 70% pour ces deux critères. La figure 2.9 montre un exemple de résultats obtenus avec les différentes mesures. Les deux premières images correspondent à l'image de test et l'image de synthèse la plus adaptée. Les quatre autres images sont celles sélectionnées par les différentes mesures de ressemblance testées.

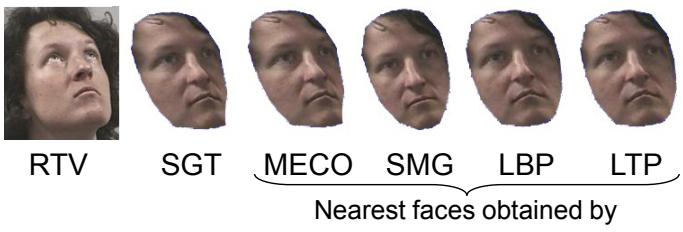


FIGURE 2.9 – Exemple d'un résultat obtenu avec les différentes mesures de ressemblance.

2.5 Limites et perspectives

Temps de calcul Il est proportionnel au nombre d'exemples dans la base de synthèse ce qui peut être un problème si on souhaite modéliser de grandes variations d'illumination, de pose et d'expression. On pourrait imaginer une structuration de la base de données sous la forme d'un arbre par exemple [Nister et Stewenius, 2006].

Limites du modèle de synthèse Le modèle de synthèse utilisé présente des limites géométriques et de textures. En effet, la texture est extraite d'une seule image frontale de la personne. Si l'éclairage introduit un artéfact (une tache spéculaire ou une surexposition d'un côté de l'image par exemple), la texture sera fausse et pourra donner de mauvais résultats. Une solution à ce problème consiste à prendre plusieurs images et à les combiner pour construire la texture à la manière d'un Modèle Actif d'Apparence [Edwards *et al.*, 1998] (AAM de l'anglais Active Appearance Model) par exemple. Les limites sont également géométriques ; le modèle n'utilise qu'une centaine de points et le pouvoir d'expression est relativement faible. Par ailleurs, Candide ne modélise pas la tête mais uniquement le visage. Le nombre de pixels dans les images de profil est donc insuffisant. Nous devons peut-être envisager l'utilisation d'un autre modèle de tête plus sophistiqué.

Précision En théorie, il suffit de diminuer le pas d'échantillonnage des poses de la base de données pour augmenter la précision ; pour des variations de pose de quelques degrés, la mesure de ressemblance n'est plus suffisamment discriminante et les variations d'expression et d'illumination deviennent prépondérantes.

Localisation de la tête Notre mesure de ressemblance est sensible aux erreurs sur la position et le facteur d'échelle du rectangle englobant produit par le détecteur de visage. Nous pourrions envisager une minimisation de la mesure de ressemblance en fonction des paramètres de translation et de changement d'échelle.

2.6 Conclusion

Nous avons présenté une méthode d'estimation grossière de pose par modèle d'apparence qui s'appuie sur une comparaison avec des visages de synthèse. L'intérêt principal de cette méthode est qu'elle ne nécessite ni phase d'apprentissage, ni base de données. Nous avons proposé une méthode itérative d'extraction d'un modèle de synthèse à partir d'une image frontale de visage dont les points caractéristiques ont été marqués. Nous avons également proposé une mesure de ressemblance inspirée de la distance de Hausdorff sur les contours orientés. Elle a été comparée avec deux autres mesures de ressemblance et a obtenu de meilleurs résultats sur les échantillons testés.

Toutefois, cette approche a montré des limites importantes qui nous ont amenés à reconsidérer l'utilisation d'une base de données pour apprendre la relation entre l'apparence d'un visage et sa pose. Nous présentons donc, dans le chapitre suivant, une approche par régression non linéaire d'estimation de la pose du visage.

Chapitre 3

Estimation de la pose par régression non linéaire

Nous proposons une méthode pour apprendre la relation entre l'apparence d'un visage et sa pose. Pour y parvenir, on dispose d'une base d'images. Chaque instance de la base est décrite par un vecteur de caractéristiques (ou attributs) et un label correspondant à la pose. La tâche de l'algorithme est d'apprendre à un outil de prédiction, le *régresseur*, la relation entre les attributs extraits de l'image et la pose de la tête. On peut utiliser directement les niveaux de gris des pixels pour caractériser l'apparence des visages. On parle alors de codage rétinien. Toutefois, le pixel n'est pas forcément le meilleur niveau de représentation de l'information contenue dans l'image. De nombreuses méthodes en analyse de visages utilisent des descripteurs d'image différents tels que les ondelettes de Haar [Viola et Jones, 2004], les ondelettes de Gabor [Wiskott *et al.*, 1997], les motifs locaux binaires [Zhao et Pietikäinen, 2007] ou les champs récepteurs gaussiens [Gourier *et al.*, 2004b]. Le nombre de descripteurs potentiels est considérable. Pris séparément, le pouvoir prédictif de chaque descripteur peut être très faible ; notre problème consiste alors à trouver la combinaison des descripteurs adaptée pour :

- *La tâche à effectuer* : dans le cadre d'une tâche de détection de visages par exemple, on cherchera des caractéristiques qui encodent des propriétés communes à tous les visages, alors que pour une application d'identification, on s'intéressera aux descripteurs qui présentent le plus de variabilité d'un individu à l'autre.
- *L'outil de prédiction utilisé* : il est peu probable que l'ensemble optimal des descripteurs pour un réseau de neurones soit optimal pour un arbre de régression. Le processus de sélection des descripteurs doit prendre en compte les spécificités du prédicteur.

3.1 Processus de sélection des descripteurs

Dans cette partie, nous abordons les quatre étapes principales du processus global de sélection de descripteurs suivantes (*cf.* figure 3.1) :

1. Sélection d'un sous-ensemble $\mathcal{FS}$ de descripteurs à évaluer parmi l'ensemble des descripteurs disponibles $\mathcal{F}$.
2. Evaluation du sous-ensemble sélectionné.

3. Evaluation du critère d'arrêt.
4. Prise en compte des performances obtenues lors de l'évaluation pour guider le nouveaux choix du sous-ensemble de descripteurs (étape 1).

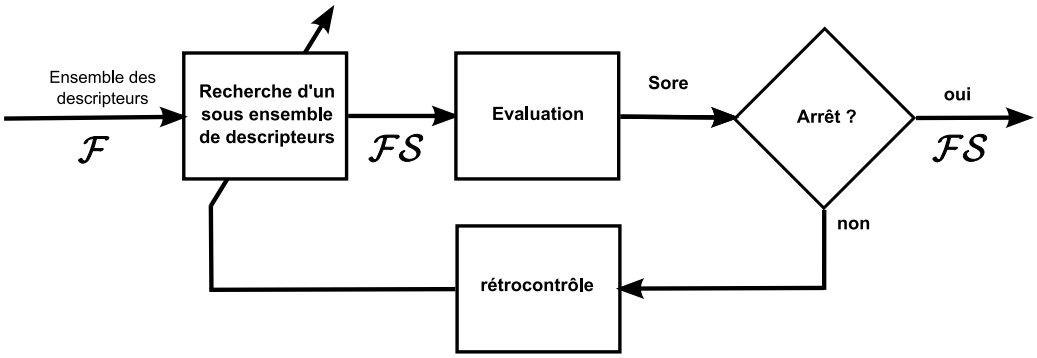


FIGURE 3.1 – Etapes du processus de sélection de descripteurs (version modifiée de la figure proposée par Liu et Yu [2005]).

3.1.1 Recherche d'un sous-ensemble de descripteurs

Lors de cette étape, l'algorithme cherche dans l'espace des descripteurs un sous-ensemble à évaluer. Les stratégies de recherche peuvent être regroupées suivant trois catégories : complète, stochastique et séquentielle.

Recherche complète. Pour un ensemble constitué de N descripteurs, il existe 2^N sous-ensembles possibles. Il n'est donc pas envisageable de faire une recherche exhaustive d'un sous-ensemble. Toutefois, différentes heuristiques d'optimisation combinatoire ont été proposées pour limiter l'espace de recherche. Somol *et al.* [2004] ont recours à l'algorithme par séparation et évaluation (*Branch and Bound* en anglais) par exemple.

Recherche stochastique. Les méthodes stochastiques sont bien adaptées à ce type de problèmes puisqu'elles permettent d'explorer un vaste espace en un temps raisonnable. Une solution simple proposée par Liu et Setiono [1996] s'appuie sur l'algorithme de Las Vegas. Elle consiste à générer aléatoirement un sous-ensemble et à l'évaluer. S'il obtient un meilleur score pour un nombre plus restreint de descripteurs que le meilleur ensemble précédemment sélectionné, il devient le meilleur ensemble. Le nombre d'itérations de ce processus est fixé préalablement par l'utilisateur. Dans cette méthode, seule la performance de l'ensemble des descripteurs est conservée d'une itération sur l'autre. Dans de nombreuses méthodes qui s'appuient sur les algorithmes génétiques [Siedlecki et Sklansky, 1989; Vafaie et DeJong, 1993; Yuan *et al.*, 1999; Farmer *et al.*, 2004] un individu est une chaîne binaire dont la taille correspond au nombre de descripteurs potentiels. Chaque bit est associé à un descripteur ; un bit à '1' signifie que le descripteur est inclus dans le sous-ensemble à

tester. A chaque itération les meilleurs individus de la population sont sélectionnés et sont utilisés pour produire les individus de la nouvelle génération (descendants). Deux opérateurs génétiques sont utilisés, le croisement et la mutation. Le croisement consiste à produire un nouvel individu à partir d'attributs (la liste des descripteurs) sélectionnés aléatoirement sur deux des meilleurs individus. La mutation modifie aléatoirement un ou plusieurs attributs d'un individu. Cette stratégie permet d'explorer aléatoirement un vaste espace de recherche tout en tenant compte des performances obtenues à chaque itération. Le recuit simulé est également une méthode d'optimisation stochastique couramment utilisée [Lin *et al.*, 2008; Meiria et Zahavi, 2006] proche des algorithmes génétiques.

Recherche séquentielle. De nombreuses variantes des méthodes de *hill-climbing* ont été proposées. Les approches *Sequential Forward Selection* sont souvent employées [Xiao *et al.*, 2009]. A chaque itération, on incorpore le descripteur qui optimise la fonction d'évaluation. L'utilisation de l'algorithme AbaBoost [Freund et Schapire, 1997] par Viola et Jones [2004] appartient à cette catégorie. A chaque étape, on sélectionne le descripteur et son classifieur associé (appelé classifieur faible) qui minimise l'erreur de prédiction sur une base de données en tenant compte des performances obtenues aux itérations précédentes. La classifieur final (appelé classifieur fort) est une combinaison linéaire des sorties des classifieurs faibles. A l'inverse, les méthodes *Sequential Backward Elimination* procèdent en retirant itérativement un descripteur de l'ensemble initial. Il n'est parfois pas possible d'optimiser le score avec un seul descripteur et l'algorithme reste prisonnier d'un minimum local. De nombreuses solutions ont été envisagées pour limiter ce problème : certaines ajoutent ou retranchent plusieurs descripteurs à la fois (*generalized sequential forward/bacward selection*), d'autres en ajoutent L puis en enlèvent R (*plus L take away R selection*). Dans les méthodes *floating search* L et R varient à chaque itération [Pudil *et al.*, 1994; Nakariyakul et Casasent, 2009].

3.1.2 Evaluation du sous-ensemble

Quelle que soit la stratégie de recherche envisagée, chaque sous-ensemble généré doit être évalué par un critère. On distingue souvent trois catégories de méthodes pour l'évaluation [Guyon et Elisseeff, 2003; Liu et Yu, 2005] : les approches *filter* (par filtrage en français), *wrapper* (par emballage en français) et les approches hybrides.

Les approches *filter* dissocient l'évaluation du sous-ensemble de l'outil de prédiction. Seules les caractéristiques intrinsèques de l'ensemble d'apprentissage sont considérées. Des mesures simples telles que l'information mutuelle ou la corrélation permettent de capturer des dépendances entre les valeurs pour un descripteur et la sortie désirée. D'autres mesures plus sophistiquées prennent en considération la redondance d'informations entre les descripteurs [Kwak et Choi, 1999; Peng *et al.*, 2005]. Cette approche est très efficace en temps de calcul, mais elle présente un inconvénient : le régresseur n'est pas intégré dans le processus de sélection [Kohavi et John, 1997].

Les approches *wrapper* utlisent les performances de l'outil de prédiction pour quantifier la pertinence du sous-ensemble de descripteurs sélectionnés. Les résultats obtenus sont souvent meilleurs puisque le sous-ensemble ainsi sélectionné est adapté à l'outil de prédiction. Cette approche est envisageable lorsque la complexité de l'algorithme d'apprentissage de l'outil de prédiction est faible. Imaginons à présent une méthode de type *Sequential Forward Selection* appliquée à un outil de prédiction plus complexe tel qu'un réseau de neurones. A chaque nouvelle itération, il faudrait tester chaque descripteur comme nouvelle entrée du réseau. Sélectionner M descripteurs parmi N nécessite environ $N * M$ apprentissages. Cette solution n'est donc pas viable lorsque le nombre de descripteurs potentiels est trop important. Certaines solutions ont été proposées pour répondre à ce problème. Leyrit *et al.* [2008] par exemple, proposent d'utiliser AdaBoost pour sélectionner les classifieurs faibles, puis d'entraîner un séparateur à vaste marge (ou SVM de l'anglais *Support Vector Machine*) à partir des sorties de ces classifieurs. Toutefois, rien ne garantit que les descripteurs sélectionnés par AdaBoost soient pertinents pour un autre outil de prédiction.

Les approches hybrides ont aussi été proposées [Van Dijck et Van Hulle, 2006; Xing *et al.*, 2001; Das, 2001; Yuan *et al.*, 1999] afin de combiner les performances des approches *wrapper* à la rapidité de calcul des approches *filter*. Dans [Yuan *et al.*, 1999], un algorithme génétique sélectionne un sous-ensemble de descripteurs en minimisant un critère d'incohérence (*inconsistency criterion*). Deux instances sont incohérentes lorsque les vecteurs d'entrée (les valeurs des descripteurs) sont identiques pour des sorties à prédire différentes [Almuallim et Dietterich, 1994]. Le sous-ensemble sélectionné est ensuite élagué par un algorithme de *Sequential Backward Elimination*. Van Dijck et Van Hulle [2006] proposent une première étape de filtrage pour sélectionner des descripteurs utiles. Ils utilisent l'information mutuelle entre les descripteurs et la sortie désirée pour sélectionner les descripteurs pertinents et l'information mutuelle entre descripteurs pour identifier les redondances. Le choix de descripteurs est ensuite affiné à l'aide d'un algorithme génétique qui sélectionne des entrées adaptées pour un réseau de neurones.

3.1.3 Critère d'arrêt

L'algorithme de sélection de descripteurs doit définir les conditions d'arrêt du processus. La plupart des méthodes suivent l'une de ces conditions [Liu et Yu, 2005] :

- L'ensemble des descripteurs a été entièrement exploré.
- Une des limites a été atteinte (nombre maximum d'itérations ou de descripteurs par exemple).
- L'ajout ou la suppression de nouveaux descripteurs n'améliore pas les résultats.
- Les performances obtenues sont suffisantes pour la tâche envisagée.

A mesure que l'on ajoute de nouveaux descripteurs, la dimension de l'espace augmente. Les méthodes peuvent donc être sensibles au sur-apprentissage, en particulier s'il y a peu

d'éléments dans l'ensemble d'apprentissage par rapport à la dimension du vecteur de caractéristiques. On envisage donc souvent l'utilisation d'une base de cross-validation. Une autre stratégie couramment proposée [Bishop, 1995; Xing *et al.*, 2001; Zhou *et al.*, 2005] consiste à introduire un terme de régularisation dans la fonction de coût à optimiser. Les méthodes de sélection de modèles qui s'appuient sur un critère analytique tels que AIC (*An Information Criterion* de Akaike 1974) ou BIC (*Bayesian Information Criterion* de Schwarz 1978), cherchent le meilleur compromis (au sens de ce critère) entre l'erreur d'estimation et la complexité du modèle.

3.1.4 Boucle de rétrocontrôle

Le rôle de la boucle de rétrocontrôle est d'utiliser les informations de performance obtenues lors de la phase d'évaluation pour orienter le choix des futurs descripteurs. Le rétrocontrôle peut prendre des formes très variées. Dans le cas d'une recherche par algorithme génétique par exemple, l'évaluation permet de sélectionner les meilleurs individus de la population et de privilégier l'exploration dans le voisinage de ces individus. Dans une approche de sélection par *boosting*, les performances de l'outil de prédiction pondèrent l'impact des exemples dans le choix du descripteur suivant. Si un exemple qui a été mal prédit avec l'ensemble des descripteurs sélectionnés, il devrait avoir plus d'importance dans le choix du descripteur suivant. L'algorithme aura ainsi tendance à sélectionner des descripteurs complémentaires.

3.2 Méthodes de *Boosting* pour la régression

Le *boosting* regroupe un ensemble de méthodes récentes d'apprentissage statistique pour la classification et la régression [Freund et Schapire, 1997]. L'idée directrice est d'estimer une valeur de sortie y en combinant la sortie de prédicteurs faibles h_k, c'est-à-dire de prédicteurs qui sont individuellement un peu meilleurs que le hasard. Dans sa forme la plus simple, un prédicteur faible peut être assimilé à un descripteur comme dans [Cristinacce et Cootes, 2007] par exemple. L'algorithme de *boosting* pourra alors être vu comme un processus de sélection de caractéristiques. Le *boosting* sélectionne successivement chaque prédicteur faible le plus performant pour une distribution particulière des poids sur les exemples d'apprentissage. A chaque itération, la distribution des poids sur des exemples est calculée en fonction de l'erreur aux itérations précédentes. AdaBoost (pour *adaptive boosting*) [Freund et Schapire, 1997] est le premier algorithme de *boosting* qui introduit une modification des poids associés aux exemples pour entraîner et sélectionner chaque prédicteur.

Le *boosting* a été initialement proposé pour répondre à des problèmes de classification mais quelques méthodes ont été proposées pour les problèmes de régression [Shrestha et Solomatine, 2006]. Nous nous intéresserons plus particulièrement à cette deuxième catégorie d'algorithmes. Le schéma 3.2 et l'algorithme 3.3 présentent les grandes étapes des

méthodes de *boosting* pour la régression. Les principales différences entre les algorithmes de la littérature concernent les points suivants.

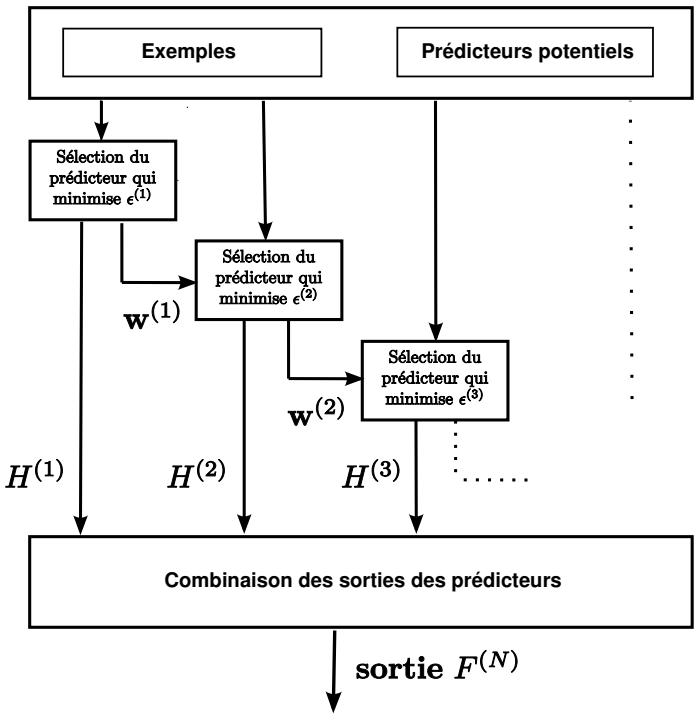


FIGURE 3.2 – Diagramme des méthodes de *boosting*

La fonction de coût ρ. La fonction de coût la plus souvent employée en régression est l'erreur quadratique $\rho_{L_2}(y, F(\mathbf{x})) = (y - F(\mathbf{x}))^2$ [Buhlmann et Yu, 2003; Friedman, 2001], avec y le label associé à un exemple et $F(\mathbf{x})$ sa valeur estimée. Drucker [1997] propose d'utiliser au choix, une fonction linéaire ou exponentielle de l'erreur absolue de prédiction. Cette dernière est également reprise par Zemel et Pitassi [2001]. M_TreeBoost [Friedman, 2001] utilise la fonction de coût de Huber issue des statistiques robustes [Huber, 2004].

Modification de la distribution des poids sur les exemples (*leveraging*). L'objectif de la repondération est d'augmenter l'importance des exemples ayant conduit à de mauvaises estimations, afin que l'algorithme se concentre sur ces exemples difficiles. Le poids $w_i^{(t)}$ associé à un exemple i à l'itération t est donc naturellement lié à l'erreur $\varepsilon_i^{(t-1)}$ de prédiction et à sa valeur à l'itération précédente $w_i^{(t-1)}$. De nombreuses fonctions empiriques ont été proposées. Dans AdaBoost.R2 par exemple, la relation entre le nouveau poids et l'erreur de prédiction est exponentielle comme pour AdaBoost. Avnimelech et Intrator [1999] puis Shrestha et Solomatine [2006] utilisent un seuil associé à l'erreur de prédiction pour différencier les exemples bien estimés des exemples avec une grande erreur de prédiction. De cette manière, ils se rapportent à un problème de classification binaire bien maîtrisé.

Le seuil doit toutefois être réglé manuellement. Dans les approches telles que GentleBoost [Friedman, 2001] et SquareLev.R [Duffy et Helmbold, 2002], la distribution des poids reste constante.

Modification des valeurs à prédire. On distingue deux catégories de méthodes. La première [Zemel et Pitassi, 2001; Drucker, 1997; Shrestha et Solomatine, 2006] suit un schéma classique de *boosting* en gardant la valeur cible constante pour chaque prédicteur à sélectionner. Les méthodes de la seconde catégorie ([Friedman, 2001; Duffy et Helmbold, 2002; Rätsch *et al.*, 2000]) ne se limitent pas à une modification des poids sur les exemples, mais modifient également les valeurs cibles y_i. Elles s'apparentent aux *forward stage-wise additive models* de Hastie et Tibshirani [1990], qui sélectionnent séquentiellement les prédicteurs pour réduire l'erreur résiduelle. Ces approches se différencient des méthodes classiques de *boosting* car la distribution des poids sur les exemples n'est plus utilisée pour guider la sélection de nouveaux prédicteurs et chaque prédicteur n'apprend pas la même fonction [Zemel et Pitassi, 2001].

Combinaison des prédicteurs. La combinaison la plus souvent employée [Shrestha et Solomatine, 2006; Zemel et Pitassi, 2001] est la moyenne des réponses des prédicteurs pondérée par un coefficient qui dépend de l'erreur associée au prédicteur. Drucker [1997] utilise la médiane pondérée pour combiner les prédictions. La sortie des prédicteurs forts de type *forward stage-wise additive models* est une somme pondérée des prédicteurs faibles.

3.3 Algorithme BISAR

Nous proposons une méthode hybride *filter/wrapper* [Bailly et Milgram, 2009b,a] pour sélectionner les descripteurs d'image et apprendre la relation entre l'apparence d'un visage et sa pose. Nos principales contributions sont :

- Le critère fonctionnel flou (FFC de l'anglais *Fuzzy Functional Criterion*) : un nouveau filtre utilisé pour sélectionner les *descripteurs pertinents*.
- Une nouvelle stratégie de *boosting* qui sélectionne itérativement de nouvelles entrées complémentaires pour le régresseur

A chaque exemple $\mathbf{x}_i$ est associée une valeur de sortie y_i que l'on souhaite prédire. Les données sont séparées en un ensemble d'apprentissage, $\mathcal{A}$, de validation $\mathcal{V}$, et de test $\mathcal{T}$. Soit $\mathcal{F}$ un ensemble de descripteurs H_k $(1 \leq k \leq \mathrm{N})$ que l'on peut calculer pour chaque $\mathbf{x}_i$ tel que $h_{k,i} = H_k(\mathbf{x}_i)$. $\mathcal{F}$ peut être très grand, plus de 10 000 éléments dans notre cas. L'objectif de notre méthode est de sélectionner un sous-ensemble de descripteurs $\mathcal{FS} \subset \mathcal{F}$ adapté à un régresseur spécifique.

A l'initialisation, l'ensemble $\mathcal{FS}$ est vide. Nous sélectionnons un descripteur à l'aide du critère fonctionnel flou que nous détaillerons dans une prochaine section (*cf.* 3.4.3). Ce descripteur est ajouté à $\mathcal{FS}$ et, par conséquent, utilisé comme entrée d'un régresseur. Il est

1. **Entrées** :
   - Séquence de M exemples $(\mathbf{x}_i, y_i)$
   - Ensemble de prédicteurs faibles H_k
2. **Sortie** :
   - Le prédicteur fort $F^{(T)}(\mathbf{x}_i)$
3. **Initialisation** : $t = 0$
   - Initialisation des poids sur les exemples $w_i^{(0)}$
   - Sorties à prédire $\{u_i = y_i\}_{i=1\ldots M}$
4. **Iterations** : $t = 1 \ldots T$
   - Apprendre (optionnel) et sélectioner le prédicteur $H_k^{(t)}$ qui minimise l'erreur ε^t pour la fonction de coût $\rho(u_i, F_T(\mathbf{x}_i))$ pour tous les exemples :
     $$\varepsilon^{(t)} = \sum_{i=1}^{M} w_i^{(t-1)} \rho(u_i, F_T(\mathbf{x}_i))$$
   - Mise à jour de la distribution des poids sur les exemples :
     $$w_i^{(t)} = W(\varepsilon_i^{(t)}, w_i^{(t-1)})$$
   - Mise à jour des valeurs de sortie à prédire par exemple :
     $$u_i^{(t)} = y_i - F^{(T-1)}(\mathbf{x}_i)$$
     $$\text{ou } u_i^{(t)} = y_i$$
5. Le prédicteur fort $F^{(T)}(\mathbf{x}_i)$ est une combinaison des sorties de prédicteurs faibles

FIGURE 3.3 – Algorithme général de *boosting* pour la régression

le meilleur parmi tous les descripteurs de $\mathcal{F}$ au sens du FFC sur l'ensemble d'apprentissage $\mathcal{A}$, mais l'erreur en sortie du régresseur est importante parce que l'information fournie par un descripteur de bas niveau ne permet pas de prédire correctement la sortie. Cette erreur est alors utilisée au travers d'un processus de *boosting* pour modifier l'impact de chaque exemple dans le choix du descripteur suivant. Le poids d'un exemple dont la sortie est mal estimée sera augmenté et inversement, un exemple avec une faible erreur de prédiction verra son poids diminuer. Le FFC prendra le poids de chaque exemple de la base d'apprentissage en compte à l'itération suivante et sélectionnera, par conséquent, un descripteur *complémentaire*. Le régresseur sera ensuite entraîné en utilisant les deux descripteurs de $\mathcal{FS}$ et le processus est répété jusqu'à ce qu'il n'y ait plus d'amélioration sur la base de validation $\mathcal{V}$.

Notre méthode diffère principalement d'une méthode classique de *boosting* sur deux points :

- La combinaison des descripteurs est apprise à chaque itération et ne dépend pas de l'erreur aux itérations précédentes
- On ne cherche pas directement un descripteur qui minimise une fonction de coût mais un descripteur qui minimise un critère indépendant du régresseur final en tenant compte de la distribution des poids sur les exemples

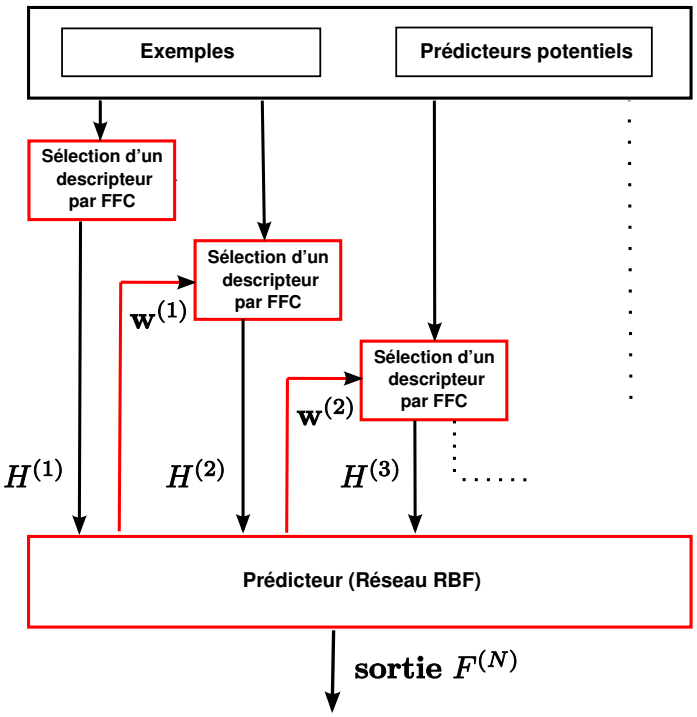


FIGURE 3.4 – Diagramme de notre méthode BISAR. Les parties en rouge sur le schéma correpondent aux parties qui diffèrent d'un algorithme de *boosting* classique.

3.4 Descripteurs d'image et prétraitements

Les images de visages sont toutes redimensionnées pour donner des vignettes de 32×32 pixels, correspondant à peu près aux dimensions utilisées par d'autres méthodes globales. Les images en couleur sont également converties en images noir et blanc et l'histogramme est ajusté dynamiquement.

Nous utilisons les filtres de Haar illustrés dans la figure 3.6 comme descripteurs d'image. Nous avons choisi ces descripteurs car ils ont donné de très bons résultats dans des domaines connexes d'analyse de visages tels que la détection [Viola et Jones, 2004] et l'alignement [Wu *et al.*, 2008]. Ils sont par ailleurs simples et très rapides à calculer à l'aide de l'image intégrale [Viola et Jones, 2004]. Nous utilisons également un autre type de descripteurs correspondant à la différence entre la somme de pixels de deux régions rectangulaires non contiguës. Il s'agit d'un sous-ensemble des descripteurs proposés par Li et Zhang [2004]. Les descripteurs sont paramétrés par quatre valeurs, x_1, y_1, dx et dy. Les valeurs x_1 et y_1 correspondent respectivement à la position horizontale et verticale du descripteur dans la vignette du visage. dx et dy sont la hauteur et la largeur d'un des rectangles (*cf.* figure 3.6). Deux paramètres supplémentaires x_2 et y_2 sont nécessaires pour définir la position du second rectangle des descripteurs de Li et Zhang [2004].

1. **Entrées** :
   - Séquence de M exemples $(\mathbf{x}_i, y_i)$
   - Ensemble de N prédicteurs faibles H_k
2. **Sortie** :
   - Le régresseur $Reg(\mathbf{x}_i)$
3. **Initialisation** : $t = 0$
   - Initialisation des poids $\mathbf{w}^{(0)} = (\frac{1}{M}, \ldots, \frac{1}{M})$ et $FS = \emptyset$
   - Calcul de Q utilisé dans l'estimation du FFC (*cf.* section 3.4.3) :
   $$Q_i = \frac{1}{N^2} \sum_j L_a(|h_{k,i} - h_{k,j}|).(1 - L_b(|y_i - y_j|))$$
4. **Iteration** : $t = 1 \ldots T_{max}$
   - Evaluation du critère FFC pour chaque descripteur :
   $$FFC(H_k) = \sum_i w_i^{(t)} Q_i$$
   $$H^{(t)} = \arg\min_{H_k \in \mathcal{FS}} FFC(H_k)$$
   - Ajout du meilleur descripteur $H^{(t)}$ à l'ensemble des descripteurs sélectionnés. $\mathcal{FS} \leftarrow \mathcal{FS} \cup H^{(t)}$ et $\mathcal{F} \leftarrow \mathcal{F} \backslash H^{(t)}$.
   - Apprentissage de Reg_t avec $(h_i^{(1)}, \ldots, h_i^{(t)})$ l'entrée et y_i la sortie désirée pour chaque exemple $\mathbf{x}_i$.
   - Estimation de l'erreur $\varepsilon^{(t)}$
   $$\varepsilon^{(t)} = \sum_i |y_i - Reg_t(\mathbf{x}_i)|$$
   - Mise à jour des poids en utilisant une des stratégies de *boosting* proposées et l'erreur $\varepsilon^{(t)}$ (*cf.* 3.6)
5. **Le régressseur** sélectionné au final est celui ayant obtenu la plus faible erreur de prédiction sur l'ensemble de validation $\mathcal{V}$ au cours des T_{max} itérations.

FIGURE 3.5 – Algorithme BISAR

3.4.1 Critère de sélection

L'algorithme BISAR nécessite un filtre pour sélectionner les descripteurs. Ce critère doit être capable de mesurer la dépendance fonctionnelle entre un descripteur et la sortie à prédire. On parle de dépendance fonctionnelle s'il existe une relation univoque qui associe à toute valeur du descripeur, une valeur de sortie unique. De plus, cette mesure doit être capable de prendre en compte des poids sur les exemples d'apprentissage. Nous présenterons dans un premier temps l'information mutuelle (section 3.4.2) qui est très souvent utilisée pour sélectionner des descripteurs. Dans un second temps, nous détaillerons le nouveau critère fonctionnel flou (section 3.4.3).

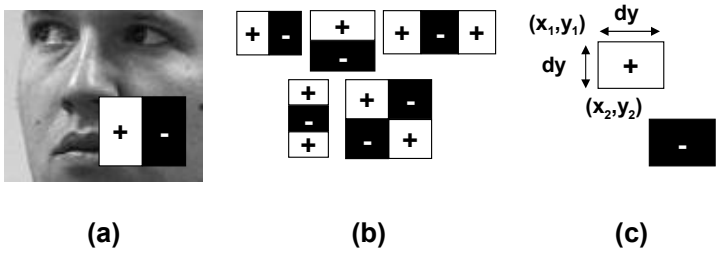


FIGURE 3.6 – Descripteurs d'image. (a) Un exemple de descripteur associé à une vignette de visage. (b) Représentation des filtres de Haar. (c) Illustration des filtres proposés par Li et Zhang [2004]

3.4.2 Entropie et information mutuelle

L'entropie mesure l'incertitude d'une variable aléatoire. Elle a été introduite par Shannon dans l'étude des canaux de communication. L'entropie d'une variable aléatoire discrète X à valeurs dans $\mathcal{X}$ est définie par :

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{3.1}$$

$p(x)$ est la probabilité, $p(x) = Pr\{X = x\}, x \in \mathcal{X}$.

L'entropie restante provenant de la variable aléatoire Y, si l'on connaît parfaitement la seconde variable aléatoire X, est donnée par l'entropie conditionnelle :

$$H(Y|X) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x) \tag{3.2}$$

On peut alors mesurer l'information mutuelle qui représente la dépendance statistique de ces deux variables aléatoires, c'est-à-dire le gain sur l'incertitude de Y quand X est connue.

$$I(X; Y) = H(Y) - H(Y|X) \tag{3.3}$$

L'information mutuelle peut être directement calculée par :

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x), p(y)} \tag{3.4}$$

Pour des variables aléatoires continues, l'information mutuelle est donnée par :

$$I(X;Y) = \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{3.5}$$

Le problème principal, ici, est d'estimer les densités de probabilité (ddp) $p(x)$, $p(y)$ et $p(x,y)$. Les approches non paramétriques sont largement utilisées lorsque l'on ne dispose d'aucune information *a priori* sur la distribution à estimer. Battiti [1994] et Kwak et Choi [2002b] approximent la *ddp* par un histogramme mais l'erreur d'estimation de l'information mutuelle est grande. Nous avons préféré utiliser les fenêtres de Parzen qui donnent de meilleurs résultats [Kwak et Choi, 2002a].

L'information mutuelle a été utilisée avec succès dans de nombreuses méthodes de sélection de descripteurs [Peng *et al.*, 2005; Fleuret, 2004]. Nous présenterons une comparaison entre l'information mutuelle et notre critère dans la partie 3.7.1. Toutefois, nous pouvons déjà noter certaines limitations théoriques de l'information mutuelle :

1. Le calcul de l'information mutuelle repose sur l'estimation de la densité de probabilité. Cette estimation est particulièrement délicate lorsque la dynamique de la distribution est élevée (zones de densité très faible et zones de densité très grande).
2. L'information mutuelle est symétrique : $I(X;Y) = I(Y;X)$. Dans notre cas on souhaite savoir si y dépend de x et pas l'inverse.
3. Dans notre algorithme, nous utilisons des poids sur les exemples pour moduler leur importance. Le critère doit être capable de les prendre en compte. L'information mutuelle devra donc être totalement recalculée pour chaque descripteur et pour chaque itération de l'algorithme.
4. Les descripteurs de l'entrée et/ou de la sortie peuvent être vectoriels. La difficulté de l'estimation de la densité de probabilité augmente exponentiellement avec la dimension de l'espace. Ce phénomène est connu sous le terme de malédiction de la dimensionnalité (*curse of dimensionality*, Bellman 1961).

3.4.3 Critère Fonctionnel Flou

L'objectif de notre Critère Fonctionnel Flou (FFC de l'anglais *Fuzzy Functional Criterion*) est de sélectionner des descripteurs permettant de prédire une valeur de sortie. Nous avons identifié deux propriétés essentielles qui caractérisent l'adéquation d'un descripteur :

1. La relation entre la valeur d'un descripteur x et la sortie à prédire y doit être « idéalement » fonctionnelle : pour toute valeur x, il existe une valeur de y et une seule. Il est intéressant de noter que l'inverse n'est pas nécessaire.
2. La relation fonctionnelle doit être « lisse » : deux valeurs de descripteurs proches doivent correspondre à des valeurs de sortie proches.

Notre critère devra donc évaluer la relation entre x et y. Soit P la proposition logique « x_1 et x_2 sont proches » et Q la proposition logique « y_1 et y_2 sont proches ». S'il existe

une relation fonctionnelle lisse entre x et y, alors l'implication logique $P \Rightarrow Q$ est vraie. En se référant à la table de vérité de l'implication logique, on trouve qu'elle est équivalente à « $\neg P$ *or* Q » et également à « $\neg(P$ *and* $\neg Q)$ ». Les valeurs x et y sont réelles ; nous utilisons donc une formulation de type logique floue qui s'appuie sur des fonctions triangulaires classiques [Zadeh, 1994] notées L définies par :

$$L_a(e) = \begin{cases} 1 - \frac{|e|}{a} & \text{si} \quad |e| < a \\ 0 & \text{sinon} \end{cases} \tag{3.6}$$

Pour quantifier le fait que la proposition « x_1 et x_2 sont proches » est vraie, nous prenons en compte la valeur retournée par $L_a(|x_1 - x_2|)$. a est la largeur de la fonction triangulaire dont le choix sera discuté dans la suite du document. Nous faisons de même pour y. L'évaluation floue Z de notre implication $P \Rightarrow Q$ s'écrit donc :

$$Z = 1 - L_a(|x_1 - x_2|)(1 - L_b(|y_1 - y_2|)) \tag{3.7}$$

Pour construire notre critère nous devons additionner le résultat de l'évaluation floue Z pour tous les quadruplets (x_1, x_2, y_1, y_2) :

$$FFC = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} L_a(|x_i - x_j|)(1 - L_b(|y_i - y_j|)) \tag{3.8}$$

La constante « 1 » et le signe « - » ont été retirés. La valeur minimale du critère correspond donc au meilleur descripteur. De plus, nous divisons le critère par M^2 afin de le borner par [0 1].

Pour l'utiliser dans notre cas, nous remplaçons x_i par $h_{k,i}$, la valeur du $k^{\text{ème}}$ descripteur H_k associée au $i^{\text{ème}}$ exemple $\mathbf{x}_i$. Par ailleurs, nous introduisons une fonction de pondération sur les exemples w, telle que $\sum_{i=1}^{N} w_i = 1$

$$FFC(H_k) = \frac{1}{M^2} \sum_{i=1}^{M} w_i \sum_{j=1}^{M} L_a(|h_{k,i} - h_{k,j}|)(1 - L_b(|y_i - y_j|)) \tag{3.9}$$

On remarque que le comportement du critère FFC correspond bien à l'implication désirée, $P \Rightarrow Q$:

- Lorsque deux valeurs de descripteurs sont proches, $L_a(|h_{k,i} - h_{k,j}|)$ tend vers 1.
- Si les sorties correspondantes sont proches, ce qui est souhaitable dans le cas d'une relation fonctionnelle lisse, $(1 - L_b(|y_i - y_j|))$ tend vers 0 et le descripteur n'est pas pénalisé.
- Par contre si les valeurs de sortie sont éloignées, $(1 - L_b(|y_i - y_j|))$ tend vers 1 et le produit $L_a(|h_{k,i} - h_{k,j}|)(1 - L_b(|y_i - y_j|))$ tend vers 1. Le descripteur est donc pénalisé puisque l'implication $P \Rightarrow Q$ n'est pas respectée.

– Lorsque les valeurs des descripteurs sont éloignées, $L_a(|h_{k,i} - h_{k,j}|)$ est nul. Donc, peu importe les valeurs de sortie, le descripteur ne sera pas pénalisé.

Nous avons ainsi défini un critère qui mesure le niveau de dépendance fonctionnelle lisse entre deux variables. Le processus de sélection des descripteurs consiste donc à minimiser ce critère.

Un aspect très intéressant de la formulation (3.9) est que le FFC n'a pas besoin d'être recalculé entièrement à chaque modification des poids sur les exemples. On peut calculer une forme intermédiaire Q_i du FFC et moduler celui-ci à chaque itération, par les nouveaux poids associés aux exemples. L'équation (3.9) peut s'écrire :

$$FFC(H_k) = \sum_{i=1}^{M} w_i Q_i \tag{3.10}$$

avec :

$$Q_i = \frac{1}{M^2} \sum_{j=1}^{M} L_a(|h_{k,i} - h_{k,j}|)(1 - L_b(|y_i - y_j|)) \tag{3.11}$$

Q peut être pré-calculé une seule fois

Ainsi le critère FFC dépasse les limites formulées au sujet de l'information mutuelle :

1. Le FFC n'a pas besoin d'estimer la densité de probabilité conjointe de x et y.
2. Le FFC n'est pas symétrique.
3. Le FFC peut être précalculé indépendamment des poids sur les exemples et évalué très rapidement lorsque les poids sont connus.
4. Le FFC peut très facilement s'appliquer à des descripteurs ou des sorties de dimension quelconque (*cf.* section 3.7.4).

3.4.4 Paramètres et normalisation du FFC

Le critère FFC nécessite l'estimation de deux paramètres a et b (équation (3.9)). Ils modulent la largeur des fonctions triangulaires de l'équation (3.6) qui évaluent l'éloignement entre deux valeurs. Ils doivent donc être sélectionnés avec soin.

Les valeurs de sortie y_i sont les mêmes pour tous les descripteurs puisqu'elles sont calculées sur les mêmes exemples. Par ailleurs, les valeurs sont bornées. On peut donc choisir b comme une proportion fixe de cet intervalle. Si la distribution n'est pas uniforme, un choix du paramètre b qui repose sur une analyse statistique des données est également envisageable (une proportion de l'écart type par exemple).

Le choix du paramètre a est plus délicat car la dynamique peut varier d'un descripteur à l'autre. Si on choisit une valeur de a différente pour chaque descripteur, ils seront difficilement comparables. Nous proposons de les normaliser en remplaçant les valeurs x des

descripteurs par leur rang. Pour N exemple, $\text{rang}(x)/N$ est toujours dans l'intervalle [0 1] et les valeurs sont régulièrement espacées. Ainsi, le critère n'est pas indépendant par rapport au paramètre a mais il est moins sensible à ce paramètre car la dynamique des descripteurs est la même pour tous.

3.5 Régresseur

3.5.1 Réseaux de neurones à fonctions radiales

Le régresseur est le deuxième élément important de notre système. Nous nous sommes intéressés à différentes architectures de réseaux RBF (de l'anglais *Radial Basis Function*). Ce sont des réseaux de neurones reconnus dans le domaine de l'interpolation [Girosi et Poggio, 1990] qui sont capables d'approximer n'importe quelle fonction [Park et Sandberg, 1991]. Un réseau RBF comporte une couche d'entrée, une couche cachée composée de neurones à fonction radiale et une couche de sortie linéaire. La figure 3.7 représente l'architecture générique d'un tel réseau.

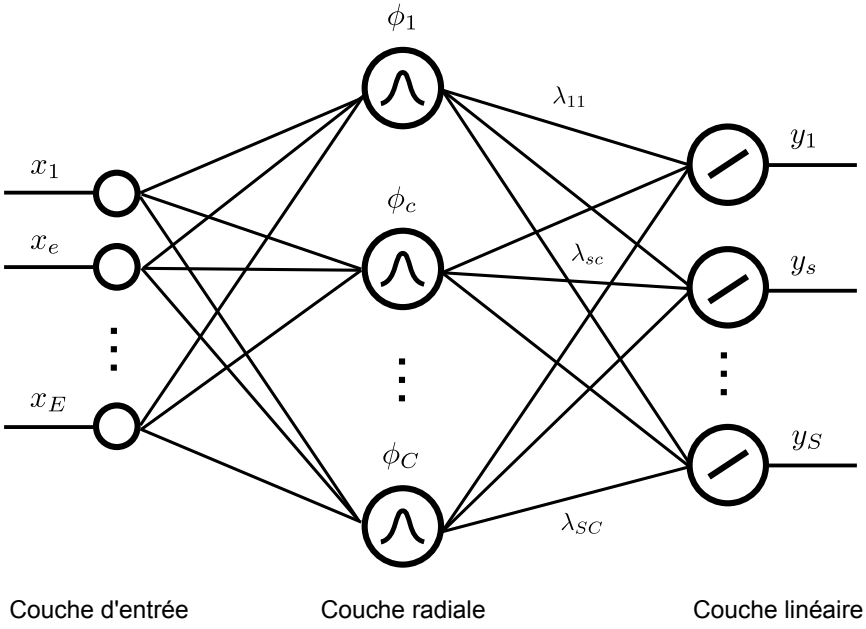


FIGURE 3.7 – Architecture d'un réseau RBF correspondant à l'équation 3.12.

La relation entre un vecteur d'entrée $\mathbf{x} = (x_1, \ldots, x_e, \ldots, x_E)$ et y_s le $s^{\text{ème}}$ neurone de sortie est donnée par :

$$y_s(\mathbf{x}) = \sum_{c=1}^{C} \lambda_{sc} \phi_c(\mathbf{x}) \tag{3.12}$$

avec C le nombre de neurones sur la couche cachée, λ_{sc}, la contribution du $c^{\text{ème}}$ neurone caché à la $s^{\text{ème}}$ sortie. La fonction d'activation du $c^{\text{ème}}$ neurone caché $\phi_c(\mathbf{x})$ est donnée par

une fonction sphérique radiale :

$$\phi_c(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_c\|^2}{2\sigma_c^2}\right) \tag{3.13}$$

avec $\boldsymbol{\mu}_c$ le centre de la fonction radiale, et σ_c sa largeur. Cette fonction radiale peut être remplacée par une fonction Gaussienne de covariance $\boldsymbol{\Sigma}_c$ quelconque :

$$\phi_c(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c (\mathbf{x} - \boldsymbol{\mu}_c)\right) \tag{3.14}$$

3.5.2 Apprentissage

L'apprentissage d'un réseau RBF se décompose en deux étapes. Une première phase non supervisée vise à estimer les paramètres des fonctions radiales ($\boldsymbol{\mu}_c$ et σ_c dans le cas de fonctions radiales ou $\boldsymbol{\Sigma}_c$ dans le cas de fonctions Gaussiennes). Pendant la seconde phase de l'apprentissage, les paramètres des gaussiennes sont figés pendant l'estimation des poids de la couche de sortie.

De nombreuses solutions ont été proposées pour l'optimisation des fonctions radiales [Bishop, 1995]. Nous avons choisi l'approche de Moody et Darken [1989] qui s'appuie sur l'algorithme de coalescence (*clustering* en anglais) des *K-means*. Il s'agit d'une procédure itérative d'estimation de la position de K centres qui reflètent au mieux la distribution des exemples d'apprentissage. Soit $\mathbf{x}_i$ un des N exemples. On souhaite trouver K vecteurs représentatifs $\boldsymbol{\mu}_c$ (prototypes), avec $c = \{1, \ldots, K\}$. L'algorithme cherche à partitionner l'ensemble des exemples $\{\mathbf{x}_i\}$, en K ensembles disjoints $\mathcal{S}_c$ de N_c exemples, de manière à minimiser :

$$J = \sum_{c=1}^{K} \sum_{i \in \mathcal{S}_c} \|\mathbf{x}_i - \boldsymbol{\mu}_c\| \tag{3.15}$$

avec $\boldsymbol{\mu}_c$ la moyenne des vecteurs d'exemple de l'ensemble $\mathcal{S}_c$

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{i \in \mathcal{S}_c} \mathbf{x}_i \tag{3.16}$$

Les K centres sont initialisés aléatoirement. A chaque itération on sélectionne pour chaque exemple $\mathbf{x}_i$ le centre $\boldsymbol{\mu}_c$ le plus proche au sens de la distance euclidienne. On forme ainsi de nouveaux ensembles $\mathcal{S}_c$ qui permettront d'estimer les nouvelles valeurs de $\boldsymbol{\mu}_c$. Le processus prend fin lorsque les centres des groupes sont stables ou lorsque le nombre maximum d'itérations est atteint. Les autres paramètres des fonctions radiales (σ_c ou $\boldsymbol{\Sigma}_c$ par exemple) sont estimés à partir des exemples $\mathbf{x}_i$ de chaque ensemble $\mathcal{S}_c$ correspondant.

Ces paramètres peuvent également être estimés par une méthode d'Espérance-Maximisation (EM) [Dempster *et al.*, 1977]. La matrice de covariance $\boldsymbol{\Sigma}_c$ est alors calculée à chaque itération. Cette approche peut introduire de l'instabilité dans le processus d'apprentissage, car la matrice de covariance est souvent mal conditionnée.

Lorsque les paramètres des fonctions radiales sont fixés, on estime les poids λ_{sc} entre la couche cachée et la couche de sortie. L'objectif est de minimiser l'erreur quadratique entre la sortie estimée par le réseau $\hat{y}_s$ et la sortie désirée y_s sur la base d'apprentissage. La relation (3.12) peut s'écrire sous forme matricielle :

$$\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{\Lambda}\boldsymbol{\phi} \tag{3.17}$$

avec $\mathbf{\Lambda}$ la matrice $S \times C$ des poids λ_{sc} et $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_C)^T$. On peut donc facilement estimer des poids par une méthode d'inversion matricielle :

$$\mathbf{\Lambda} = \mathbf{Y}\mathbf{\Phi}^{\dagger} \tag{3.18}$$

avec $\mathbf{Y}$ contenant en colonne les vecteurs des sorties désirées pour chaque exemple et $\mathbf{\Phi}$ contenant en colonne les valeurs de sortie des fonctions de la couche cachée pour chaque exemple.

3.5.3 Réseaux de neurones de régression généralisée

Nous avons également testé les réseaux de neurones de régression généralisée (GRNN de l'anglais *Generalized Regression Neural Network*) que l'on peut considérer comme une version simplifiée des réseaux RBF [Wasserman, 1993]. Chaque prototype de la couche cachée correspond à un exemple de la base d'apprentissage et les poids entre les neurones de la couche cachée et la couche de sortie sont les valeurs de sortie des exemples d'apprentissage. Cela signifie que ce réseau ne nécessite pas de phase d'apprentissage. La largeur des fonctions radiales σ_c est le seul paramètre. Il définit le degré de lissage de l'interpolation. Ce réseau est très semblable aux interpolations exactes de Powell [1987]

3.6 Stratégie de *boosting*

Le réseau de neurones débute avec une seule entrée correspondant au meilleur descripteur sélectionné par le FFC avec une distribution uniforme des poids sur les exemples. A la deuxième itération, un nouveau réseau est entraîné avec deux neurones en entrée. La nouvelle entrée correspond au meilleur descripteur sélectionné parmi tous les descripteurs (à l'exception de celui déjà sélectionné) pour une distribution des poids qui dépend de l'erreur du réseau à l'étape précédente. Les exemples dont la sortie a été mal estimée précédemment recevront un poids plus fort que les autres. On souhaite ainsi que ces nouveaux poids amènent le critère FFC à sélectionner les descripteurs adaptés pour ces exemples. Ce paradigme de *boosting* améliore ainsi le processus de régression.

Contrairement à AdaBoost [Freund et Schapire, 1997], la repondération itérative est uniquement utilisée pour sélectionner les meilleurs descripteurs complémentaires. La manière de les combiner est entièrement gérée par l'outil de prédiction. Cette stratégie de *boosting* indirect peut être vue comme une régularisation pour éviter le sur-apprentissage.

3.6.1 Fonction de repondération

Plusieurs stratégies sont possibles pour guider le processus de sélection. Nous avons testé trois stratégies de *boosting*. La première repose sur un processus sans mémoire (*memoryless* en anglais). Le poids $w_i^{(t+1)}$ du $i^{\text{ème}}$ exemple à l'itération $t+1$ dépend uniquement de l'erreur quadratique $(\varepsilon_i^{(t)})^2$ à l'itération précédente. Dans ce cas, nous avons choisi la relation :

$$w_i^{(t+1)} = \frac{(\varepsilon_i^{(t)})^2}{\sum_{i=1}^{M}(\varepsilon_i^{(t)})^2} \tag{3.19}$$

La seconde stratégie est cumulative. Le poids de chaque exemple dépend de l'erreur du régresseur et du poids à l'itération précédente :

$$\tilde{w}_i^{(t+1)} = \begin{cases} w_i^{(t)} \text{ si } \varepsilon_i^{(t)} < \underset{1<i<M}{\text{mediane}}(\varepsilon_i^{(t)}) \\ min\left\{\alpha w_i^{(t)}, w_{max}\right\} \text{ sinon} \end{cases} \tag{3.20}$$

$$w_i^{(t+1)} = \frac{\tilde{w}_i^{(t+1)}}{\sum_{i=1}^{M}\tilde{w}_i^{(t+1)}} \tag{3.21}$$

Les poids de la moitié des exemples avec la plus grande erreur de prédiction sont multipliés par un facteur d'accumulation constant α. La constante w_{max} est utilisée pour éviter que le poids de certains exemples n'augmente démesurément et entraîne un effet de surapprentissage. Dans nos expériences, α est initialisé à 1.1 et w_{max} à 0.1

La troisième fonction de repondération est la même que dans AdaBoost.R2 [Drucker, 1997]. La performance du régresseur est mesurée par la fonction de coût :

$$L_i^{(t)} = \left(\frac{\varepsilon_i^{(t)}}{\max\limits_{i=1\ldots M}\varepsilon_i^{(t)}}\right)^2 \tag{3.22}$$

Cette fonction est moyennée sur tous les exemples pondérés :

$$\bar{L}^{(t)} = \sum_{i=1}^{M} w_i^{(t)} L_i^{(t)} \tag{3.23}$$

Connaissant $\bar{L}^{(t)}$, le paramètre de mise à jour des poids $\beta^{(t)}$ est calculé par :

$$\beta^{(t)} = \bar{L}^{(t)}/(1-\bar{L}^{(t)}) \tag{3.24}$$

Finalement, les nouveaux poids sur les exemples sont calculés par :

$$\tilde{w}_i^{(t+1)} = w_i^{(t)} \left(\beta^{(t)}\right)^{(1-L_i^{(t)})} \tag{3.25}$$

et normalisés par :

$$w_i^{(t+1)} = \frac{\tilde{w}_i^{(t+1)}}{\sum_{i=1}^{M} \tilde{w}_i^{(t+1)}} \tag{3.26}$$

Contrairement aux autres algorithmes de *boosting* pour la régression, aucun paramètre n'a besoin d'être réglé, et les tests sur différents ensembles de données ont montré de bonnes performances [Shrestha et Solomatine, 2006]. Dans la suite du document, les trois stratégies de *boosting* seront référencées sous les termes de stratégie *memoryless* (3.19), *median* (3.20) et AdaBoost.R2 (3.25).

3.6.2 Critère d'arrêt

Nous avons évoqué dans la section 3.1.3 les principales conditions utilisées pour arrêter le processus de sélection des descripteurs. Dans les approches de type *wrapper* et hybrides, l'évaluation des performances de l'outil de prédiction est un critère naturel d'arrêt très souvent utilisé. Toutefois les données d'apprentissage sont bruitées. En effet, nous avons vu dans le chapitre précédent que la localisation du visage multi-orientation donnait des résultats bruités. De plus, l'étiquetage de la pose du visage est imparfait, en particulier dans la base de données Pointing 04. Les risques de sur-apprentissage sont donc importants, en particulier dans une approche par *boosting* qui risque de concentrer son attention sur les observations aberrantes. L'arrêt prématuré (ou *early stopping* en anglais) est une méthode simple et très efficace en pratique [Finnoff *et al.*, 1993]. Cette méthode consiste à diviser l'ensemble des exemples en un ensemble d'apprentissage $\mathcal{A}$ et un ensemble de validation $\mathcal{V}$. Pendant l'apprentissage sur $\mathcal{A}$ on calcule périodiquement l'erreur sur $\mathcal{V}$ et on arrête l'apprentissage lorsque l'erreur sur $\mathcal{V}$ est supérieure à la précédente évaluation. Cette approche utilise $\mathcal{V}$ pour anticiper le comportement de l'outil de prédiction sur l'ensemble de test $\mathcal{T}$. Dans la réalité, la courbe d'évolution de l'erreur en généralisation n'est pas lisse et contient de nombreux minima locaux au cours de l'apprentissage. Dans ces conditions il devient très difficile de définir un critère universel d'arrêt comme le montre l'étude de Prechelt [1998]. Nous avons choisi de sélectionner un nombre prédéfini de descripteurs (et donc d'itérations) important. On sélectionne ensuite le régresseur et le jeu de descripteurs ayant obtenu les meilleurs résultats sur la base de validation.

3.7 Résultats

Nous avons réalisé nos tests sur les bases Pointing 04 et FacePix (*cf.* section 1.4) et nous avons comparé nos résultats à ceux obtenus par les méthodes présentées dans la section 1.5 du document.

3.7.1 Evaluation des critères de sélection de caractéristiques

Dans cette première expérience, nous comparons le Critère Fonctionnel Flou et l'Information Mutuelle. Afin de réduire l'impact des paramètres extérieurs, l'outil de régression est un GRNN et aucune stratégie de *boosting* n'est mise en oeuvre, c'est-à-dire les poids sur les exemples restent constants au cours des itérations de BISAR. Le nombre optimal de descripteurs est choisi en fonction du score sur l'ensemble de validation. Les résultats reportés dans le tableau 3.1 donnent l'avantage au FFC. En présence de *boosting*, le FFC est plus rapide à calculer que l'Information Mutuelle à l'aide de l'équation (3.10). Ces deux raisons nous amènent à privilégier le FFC dans les prochaines expériences.

TABLE 3.1 – FFC vs MI : Erreur moyenne absolue en degrés sur l'ensemble de test. Les valeurs entre parenthèses indiquent le nombre de descripteurs sélectionnés.

	Pointing 04				FacePix
	Localisation Manuelle		Localisation Automatique		
	Erreur Pan	**Erreur Tilt**	**Erreur Pan**	**Erreur Tilt**	**Erreur Pan**
FFC	**8.1**° (580)	**8.4**° (560)	**12.5**° (480)	15.9° (100)	**6.2**° (191)
MI	8.4° (600)	8.9° (560)	12.8° (460)	**15.7**° (140)	6.4° (180)

3.7.2 Evaluation des stratégies de *boosting*

Dans cette partie, nous comparons différentes stratégies de *boosting*. Nous utiliserons le GRRN comme régresseur. Ainsi, aucun paramètre n'aura besoin d'être réglé et les résultats ne dépendront pas d'une initialisation aléatoire. On effectue des tests sur Pointing 04 pour des visages localisés manuellement et automatiquement. De cette manière, on peut tester la résistance au bruit et aux données aberrantes. Le tableau 3.2 présente les résultats pour les trois stratégies de *boosting*. On présente également les résultats obtenus sans *boosting* (désigné par *none* dans le tableau). Nous étudions les performances de BISAR jusqu'à 600 itérations et nous sélectionnons le régresseur avec les meilleures performances sur l'ensemble de validation.

L'effet du *boosting* est évident puisque n'importe quelle fonction de repondération améliore les résultats sans *boosting*. Par ailleurs, la stratégie de repondération d'AdaBoost.R2 semble légèrement supérieure et obtient les meilleurs résultats pour trois des cinq ensembles de données. La figure 3.7.2 illustre l'effet du *boosting* au cours des itérations. On remarque que l'erreur décroît plus rapidement pour la stratégie AdaBoost.R2 dans les premières itérations. La stratégie *median* donne de meilleurs résultats sur FacePix mais nécessite jusqu'à trois fois plus de descripteurs que les autres méthodes. La stratégie *memoryless* obtient de bons résultats et surpasse la stratégie AdaBoost.R2 sur l'estimation de l'angle pan lorsque le visage est localisé manuellement, mais tend à sur-apprendre lorsque les données sont bruitées (*cf.* figure 3.8(b)). Les résultats sont meilleurs sur FacePix que sur Pointing 04. Cela est probablement lié aux différences soulignées dans la partie 1.4.2, page 37 (uniquement

TABLE 3.2 – Erreur moyenne absolue en degrés sur l'ensemble de test pour différentes stratégies de *boosting*. Les valeurs entre parenthèses indiquent le nombre de descripteurs sélectionnés.

	Pointing 04				FacePix
	Localisation manuelle		Localisation automatique		
	Err. pan	**Err. tilt**	**Err. pan**	**Err. tilt**	**Err. pan**
None	8.1° (582)	8.4° (573)	12.5° (467)	15.9° (102)	6.2° (191)
Memoryless	**7.3**° (553)	8.5° (267)	11.6° (389)	13.5° (78)	5.4° (243)
Median	7.6° (569)	8.9° (223)	11.9° (464)	16.5° (55)	**4.5**° (599)
AdaBoost.R2	8.2° (382)	**7.6**° (457)	**10.9**° (430)	**12.3**° (120)	6.0° (265)

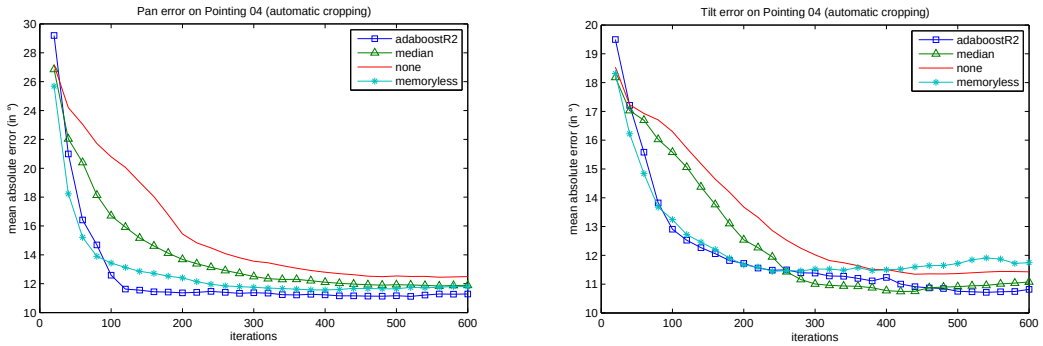
des variations suivant pan, biais dans le cadrage des visages, plus grande précision dans le processus d'acquisition de la vérité terrain).

3.7.3 Evaluation de l'architecture

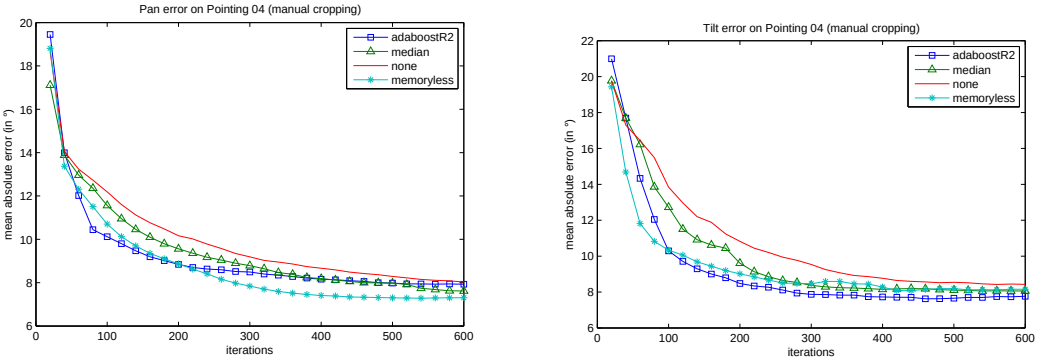
Dans cette partie, nous essayons d'autres architectures de réseaux RBF. On réalise 300 itérations pour chaque régresseur sur Pointing 04 et FacePix. On conserve la fonction de repondération AdaBoost.R2 puisqu'elle a donné les meilleurs résultats dans la section précédente. Les tests sont menés sur deux réseaux RBF qui diffèrent par le nombre de neurones sur la couche cachée. Le nombre de neurones pour RBF 1 correspond à la moitié du nombre d'exemples d'apprentissage et un cinquième pour RBF 2. Les cellules de la couche cachée sont choisies par un algorithme des *K-means* et la matrice de covariance de chaque groupe est calculée uniquement à la fin du processus. Le tableau 3.3 montre que les performances sont directement liées au nombre de cellules de la couche cachée. Bien que le nombre de cellules de la couche cachée soit très grand dans le GRNN, on n'observe pas de phénomène de sur-apprentissage. Ce phénomène est peut-être en partie lié à l'utilisation de fonctions radiales sphériques qui introduisent un biais fort. Par ailleurs, la variance du régresseur augmente avec le nombre de cellules de la couche cachée, mais le nombre restreint de descripteurs selectionnés joue un rôle de régularisation [Bühlmann et Hothorn, 2007]

TABLE 3.3 – Erreur moyenne absolue en degrés sur l'ensemble de test pour différentes architectures du régresseur. Les valeurs entre parenthèses indiquent le nombre de descripteurs sélectionnés.

	Pointing 04		FacePix
	Erreur pan	**Erreur tilt**	**Erreur pan**
GRNN	**11.4**° (288)	**12.3**° (120)	**6.0**° (265)
RBF 1	14.0° (217)	15.8° (256)	7.2° (120)
RBF 2	15.2° (161)	16.0° (284)	8.5° (237)



(a) (b)



(c) (d)

FIGURE 3.8 – Evolution de l'erreur absolue moyenne au cours des itérations pour différentes stratégies de *boosting*.

3.7.4 Apprentissage des poses séparées *vs* groupées

Dans cette expérience, les angles pan et tilt sont appris conjointement. Pour y parvenir, nous avons légèrement modifié le critère FFC. La valeur de sortie est vectorielle ; nous avons remplacé la valeur absolue des différences de sorties par la norme L_1 dans l'équation (3.9). Apprendre deux informations au lieu d'une peut sembler, *a priori*, plus difficile. Toutefois, les informations de pan et de tilt sont liées et l'apprentissage conjoint donne de meilleurs résultats sur Pointing 04 (tableau 3.4). De plus, on remarque que le nombre de descripteurs sélectionnés est moins important. On peut penser que l'apprentissage de l'angle pan induit un biais dans l'apprentissage de l'angle tilt et inversement, améliorant ainsi les performances en généralisation [Caruana, 1997].

TABLE 3.4 – Erreur moyenne absolue en degrés sur l'ensemble de test pour des apprentissages conjoints et séparés. Les valeurs entre parenthèses indiquent le nombre de descripteurs sélectionnés.

	Localisation manuelle		Localisation automatique	
	Erreur pan	**Erreur tilt**	**Erreur pan**	**Erreur tilt**
Apprentissage conjoint	**7.5°** (600)	8.0° (600)	**10.3°** (294)	**12.0°** (294)
Apprentissage séparé	8.2° (382)	**7.6°** (457)	10.9° (430)	12.3° (120)

3.8 Comparaisons avec des méthodes existantes

3.8.1 Réseau de neurones à convolution

Nous avons dans un premier temps comparé notre approche à un réseau de neurones à convolution. Les détails de l'architecture du réseau sont donnés dans la partie 1.5.1, page 38 de ce manuscrit. Les résultats obtenus sont répertoriés dans le tableau 3.5. Le CNN et BISAR présentent des résultats équivalents sur la base FacePix, mais BISAR surpasse le CNN sur la base POINTING 04.

TABLE 3.5 – Comparaison entre le réseau de neurones à convolution et BISAR

	Pointing 04		FacePix
	Erreur pan	**Erreur tilt**	**Erreur pan**
BISAR	**7.3°**	**7.6°**	4.5°
CNN	8.7°	11.5°	**4.2°**

3.8.2 Méthode de l'évaluation CLEAR 2007

Le tableau 3.6 réunit les résultats obtenus par BISAR et les trois méthodes de la campagne d'évaluation CLEAR 2006 [Stiefelhagen et Garofolo, 2007]. La section 1.5.2 page 39 décrit ces méthodes. Lorsque les visages sont localisés manuellement, les résultats de BISAR sont légèrement moins bons que ceux obtenus par Tu *et al.* [2007]. En revanche, leurs résultats chutent lorsque le visage est localisé automatiquement. Cette méthode est donc très sensible aux imprécisions de la localisation. Les mémoires autoassociatives de Gourier *et al.* [2007] obtiennent les meilleurs résultats pour l'estimation de l'angle pan, et BISAR a des résultats légèrement meilleurs pour l'angle tilt. De plus les mémoires autoassociatives sont utilisées comme classifieurs génératifs. Cette approche n'est capable d'estimer que des poses discrètes et des problèmes apparaissent lorsque le nombre de classifieurs augmente [Murphy-Chutorian et Trivedi, 2009].

BISAR obtient de meilleurs résultats que le Perceptron Multi Couche [Voit *et al.*, 2007] sur l'estimation de pan et tilt et des résultats semblables aux performances humaines reportées dans [Gourier *et al.*, 2007].

Pour chaque pose estimée, on peut chercher la valeur d'angle la plus proche dans la base de données. Si elle correspond à la vérité terrain, on considère que la réponse du régresseur est correcte. On calcule ainsi le score de classification reporté dans le tableau 3.6.

TABLE 3.6 – Résultats comparatifs des différentes méthodes sur la base Pointing 04.

Méthode	Erreur tilt	Classif.	Erreur pan	Classif.
Loc. manuelle				
BISAR	**8.0°**	65.4%	7.5°	60.6%
Tu *et al.* [2007]	8.6°	**75.7%**	**6.2°**	**72.4%**
Loc. automatique				
BISAR	**12.0°**	**56.0%**	10.2°	54.8%
Tu *et al.* [2007]	17.9°	54.8%	12.9°	49.3%
Gourier *et al.* [2007]	12.1°	53.8%	**7.3°**	**61.3%**
Voit *et al.* [2007]	12.8°	53.1%	12.3°	41.8%
Perf. humaines	9.4°	59.0%	11.8°	40.7%

3.9 Conclusion

Nous avons débuté ce chapitre par un état de l'art des méthodes de sélection de descripteurs structuré autour des quatre grandes étapes du processus de sélection. Puis nous avons focalisé notre analyse sur les méthodes de *boosting* pour la régression qui permettent de sélectionner des prédicteurs complémentaires par une stratégie de modification de la distribution des poids sur les exemples.

Nous avons ensuite présenté BISAR, notre méthode de sélection de descripteurs dédiée

aux problèmes de régression. Elle consiste à ajouter itérativement de nouvelles entrées à un réseau de neurones ; Chaque entrée est associée à un descripteur sélectionné à l'aide du critère fonctionnel flou qui mesure la dépendance fondtionnelle entre la valeur d'un descripteur et la sortie désirée. La complémentarité des descripteurs est assurée par une stratégie de *boosting*

La méthode BISAR a montré son efficacité dans le cadre de l'estimation de la pose de la tête. Elle obtient de meilleurs résultats qu'un réseau de neurones à convolution sur la base Pointing 04 et des résultats équivalent sur la base FacePix. Nous avons également obtenus des résultats du niveau de la meilleure méthode de l'évaluation CLEAR 07.

Cette étude cloture la première partie de ce manuscrit sur les méthodes globales d'estimation de la pose de la tête. Nous allons, dans la seconde partie, nous intéresser aux approches par alignement d'un modèle déformable et nous verrons comment BISAR peut-être mis à profit dans ce nouveau cadre applicatif.

Deuxième partie

Alignement d'un modèle déformable de visage

Notations

Notations relatives au modèle de forme

$\mathbf{p}$	Paramètres internes du modèle
$\mathbf{q}$	Paramètres externes du modèle
$\mathbf{b}$	Paramètres de forme $\mathbf{b} = (\mathbf{p}, \mathbf{q})^T$
$\mathbf{s}$	Modèle de forme
$\mathbf{s}_i$	Coordonnées du $i^{\text{ème}}$ point du modèle
$\bar{\mathbf{s}}$	Composante rigide du modèle de forme (également appelée forme moyenne)
$\boldsymbol{\phi}_i$	$i^{\text{ème}}$ Vecteur de déformation du modèle
$\mathbf{s}(\mathbf{b})$	Instance du modèle de forme $\mathbf{s}$ dans le repère image pour le jeu de paramètres $\mathbf{b}$
$\mathbf{s}^*_{\mathbf{I}}$	Vérité terrain des points du modèle dans l'image $\mathbf{I}$
$\hat{\mathbf{b}}_{\mathbf{I}}$	Paramètres qui ajustent au mieux le modèle à la vérité terrain dans l'image $\mathbf{I}$
$\tilde{\mathbf{b}}_{\mathbf{I}}$	Paramètres qui minimisent la fonction de coût $F(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a})$
$\mathcal{S}$	Ensemble des coordonnées des pixels définis à l'intérieur de la forme de référence $\bar{s}$
$\mathcal{B}$	Ensemble des valeurs admissibles pour le vecteur de paramètres $\mathbf{b}$
$W(\mathbf{x}; \mathbf{b})$	Fonction de transfert des coordonnées $\mathbf{x}$ de $\mathcal{S}$ dans $\mathbf{I}$ pour le jeu de paramètres $\mathbf{b}$

Notations relatives au modèle d'apparence

$\mathbf{a}$	Paramètres d'apparence du modèle
$\mathbf{g}_{mod}(\mathbf{a})$	Instance du modèle d'apparence pour le jeu de paramètres $\mathbf{a}$
$\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s})$	Vecteur d'apparence globale de la forme $\mathbf{s}$ dans l'image $\mathbf{I}$ (également appelé apparence observée ou observation).
$\mathbf{g}^{(k)}_{glob}(\mathbf{I}, \mathbf{s})$	$k^{\text{ème}}$ élément du vecteur d'apparence global.
$\mathbf{g}_{loc}(\mathbf{I}, \mathbf{x})$	Vecteur d'apparence locale dans $\mathbf{I}$ au point $\mathbf{x}$

Notations relatives à la fonction de coût

$F(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a})$	Fonction de coût globale qui mesure l'adéquation entre le modèle et l'image
$F^*(\mathbf{I}, \mathbf{s}(\mathbf{b}))$	Fonction de coût idéale
$f_i(\mathbf{I}, \mathbf{x})$	Fonction de coût locale du $i^{\text{ème}}$ point du modèle
$\mathbf{m}$	Minimum global de la fonction de coût

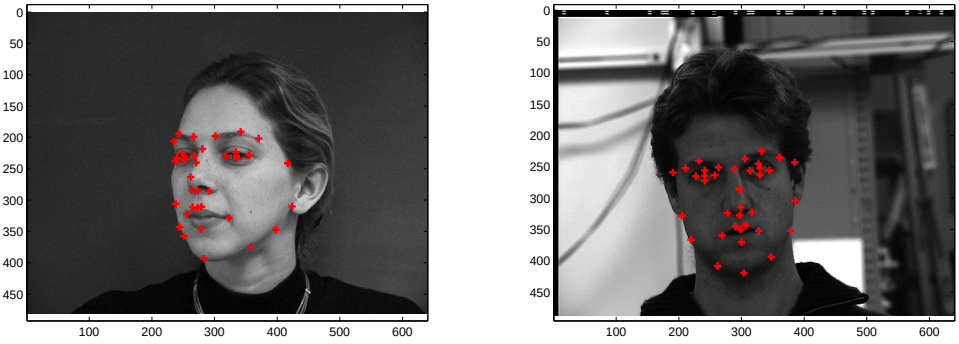
Chapitre 4

Etat de l'art

Nous présentons dans ce chapitre, un panorama des méthodes d'alignement d'un modèle de visage dans une image. Nous les analyserons au travers des quatre éléments qui les composent : le modèle de forme, le modèle d'apparence, la fonction de coût et la méthode d'optimisation.

4.1 Principe général

Pour un objet au sens large (visages, véhicules. . .), on peut définir un ensemble de points caractéristiques (également appelés amers) ; il peut s'agir par exemple des commissures de la bouche et des yeux, des extrémités des sourcils et du bout du nez (*cf.* figure 4.1). L'ensemble des amers définit un *modèle de forme* $\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_{N_{pt}}, y_{N_{pt}}) \in \mathbb{R}^{2N_{pt}}$.



(a) (b)

FIGURE 4.1 – Exemples de points caractéristiques définis sur des visages.

Un modèle peut s'entendre au sens des points physiques (3D) ou au sens des points de l'objet dans l'image. Les modifications de la forme $\mathbf{s}$, sous l'effet des différentes sources de variations géométriques (pose, expression, morphologie), sont en général décrites à l'aide d'un ensemble de paramètres $\mathbf{b}$. Ce vecteur doit contenir suffisamment d'éléments pour décrire l'ensemble des variations constatées sur l'objet.

Les deux voies principales pour construire ces modèles sont l'approche analytique où le modèle est construit manuellement, et l'approche statistique où le modèle est déduit

d'une analyse statistique des données (une ACP par exemple). Certaines composantes, dites externes, du vecteur $\mathbf{b}$ représentent des transformations géométriques (translation, rotation, changement d'échelle). Les autres composantes, dites internes, sont modifiées par la pose, les expression faciales, l'identité...

Nous allons supposer que $\mathbf{s}$ correspond à un modèle 2D de l'objet afin de décrire le processus d'alignement. Si on admet que $\mathcal{B}$ représente l'ensemble des valeurs admissibles pour le vecteur de paramètres $\mathbf{b}$ et $\mathbf{s(b)}$ la forme associée, on cherche alors à minimiser la distance entre la forme $\mathbf{s(b)}$ et la forme réelle $\mathbf{s}^*$. Cette $\mathbf{s}^*$ a en général été définie manuellement (vérité terrain). L'alignement consiste donc à trouver $\tilde{\mathbf{b}}$ tel que :

$$\tilde{\mathbf{b}} = \underset{\mathbf{b}\in\mathcal{B}}{\operatorname{argmin}}\ d\left(\mathbf{s}(\mathbf{b}), \mathbf{s}^*\right) \tag{4.1}$$

La recherche de $\tilde{\mathbf{b}}$ ne conduit pas nécessairement à une distance $d\left(\mathbf{s}(\mathbf{b}), \mathbf{s}^*\right)$ nulle car la forme réelle $\mathbf{s}^*$ n'appartient pas forcément à l'ensemble de formes $\mathcal{S}$ que l'on peut générer avec tous les $\mathbf{b}$ possibles. La position des points est recherchée **mais seule l'apparence de l'objet (l'ensemble des niveaux de gris qui représentent l'instance de l'objet dans une image) est accessible**. L'alignement consiste à établir un lien entre l'apparence et la position des points caractéristiques. On construira pour cela un *modèle d'apparence* $\mathbf{g}_{mod}$ défini par un vecteur de paramètres $\mathbf{a}$ pour caractériser les variations d'aspect de l'objet dans l'image. Une *fonction de coût* $F\left(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a}\right)$ mesurera l'adéquation entre l'image observée $\mathbf{I}$ et l'instance des modèles de forme et d'apparence. Idéalement, on souhaite que la fonction de coût reproduise le comportement de la fonction distance, c'est-à-dire que

$$\underset{\mathbf{b}}{\operatorname{argmin}}\ F\left(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a}\right) = \underset{\mathbf{b}}{\operatorname{argmin}}\ d\left(\mathbf{s}(\mathbf{b}), \mathbf{s}^*\right) \tag{4.2}$$

Une fois la fonction F connue, il reste à définir la stratégie pour trouver le meilleur jeu de paramètres. Cette étape *d'optimisation* consiste donc à trouver la meilleure position du modèle dans l'image :

$$\hat{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}}\ F\left(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a}\right) \tag{4.3}$$

On dit alors que le modèle est bien aligné (ou que le visage est correctement segmenté) si le jeu de paramètres ainsi trouvé correspond à celui estimé par (4.1), c'est-à-dire si $\hat{\mathbf{b}} = \tilde{\mathbf{b}}$. L'erreur quadratique d'alignement est donnée par $\left\|\mathbf{s}(\hat{\mathbf{b}}) - \mathbf{s}^*\right\|$.

Nous articulerons notre analyse du domaine de l'alignement de modèles autour des quatre piliers illustrés dans la figure 4.2 :

- le modèle de forme,
- le modèle d'apparence,
- la fonction de coût et
- l'algorithme d'optimisation.

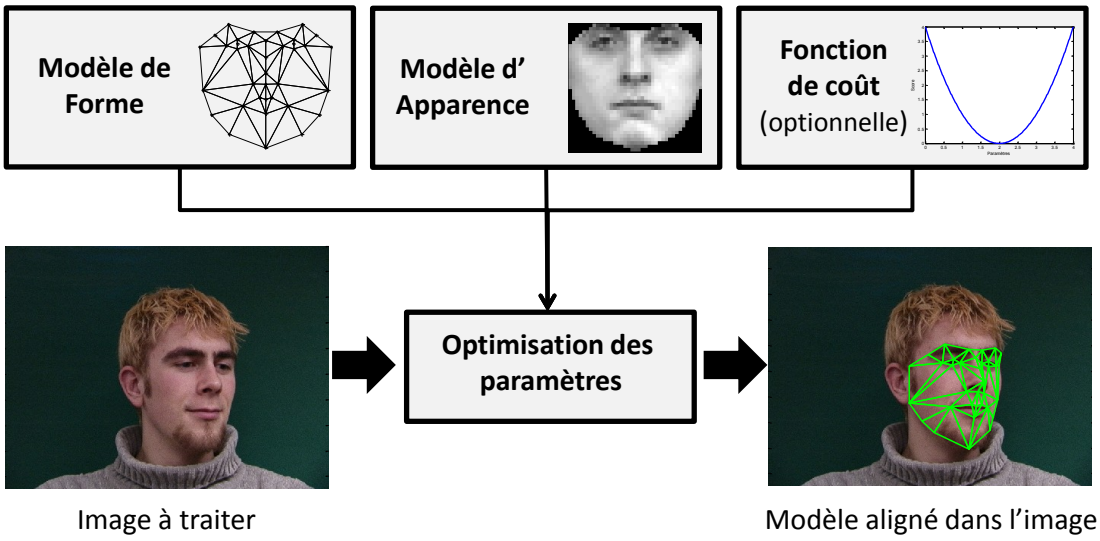


FIGURE 4.2 – Vue synthétique des différents éléments d'une méthode d'alignement.

4.2 Limites de l'état de l'art

Notre étude se bornera aux recherches dédiées à l'analyse faciale, bien que la plupart des méthodes présentées ici puissent s'étendre à d'autres domaines d'application. Par ailleurs, les autres méthodes de localisation d'éléments caractéristiques du visage sortent du cadre de cette étude. Nous n'aborderons pas, par exemple, les méthodes de détection des points caractéristiques telles que Nguyen *et al.* [2008] car elles détectent chaque point indépendamment et n'exploitent pas les contraintes géométriques entre ces points. D'autres approches telles que Senechal *et al.* [2010] et Duffner et Garcia [2005] exploitent indirectement les relations qui unissent les amers en utilisant un réseau de neurones pour les détecter conjointement. Toutefois, ces approches ne sont pas non plus abordées car elles ne font pas appel *explicitement* à un modèle de forme pour encoder les relations spatiales entre ces amers. Enfin, nous n'aborderons pas non plus les méthodes de suivi. Ce domaine est connexe à celui de l'alignement puisqu'il s'agit d'estimer les variations des paramètres d'un modèle d'une image aux images suivantes. Les résultats obtenus sont souvent visuellement impressionnants car ces méthodes exploitent :

- Une zone de recherche restreinte : les valeurs des paramètres obtenus dans les images précédentes de la séquence permettent de prédire les paramètres du modèle dans l'image courante.
- Un modèle spécifique défini à partir d'une image de la séquence (souvent la première) qui est plus facile à aligner qu'un modèle générique [Gross *et al.*, 2005]. Il est, par ailleurs, possible d'affiner le modèle de forme au cours du processus de suivi [Nguyen et Milgram, 2009].
- La cohérence temporelle qui permet de filtrer les estimations aberrantes.

De plus, elles ne peuvent pas se soustraire à une étape d'initialisation du modèle qui sera réalisée soit manuellement soit par une méthode automatique de localisation de points caractéristiques.

4.3 Modèles de forme

Un modèle de forme encode les informations purement géométriques propres à un objet. Une manière de décrire une forme est de localiser un ensemble de points caractéristiques (amers ou *landmarks*). Ces points doivent avoir une définition univoque, afin qu'un expert puisse les localiser précisément dans une image. Dryden et Mardia [1998] distinguent trois catégories de points :

- *Les points anatomiques* sont des points qui ont une définition biologique partagée par l'ensemble des organismes d'une population (la pupille de l'œil par exemple).
- *Les points mathématiques* sont localisés à partir d'une propriété mathématique ou géométrique (un coin par exemple).
- *Les pseudo-points caractéristiques* sont définis par rapport aux points anatomiques ou mathématiques. Typiquement, il s'agit de points uniformément répartis le long d'un contour entre deux points anatomiques afin d'obtenir une représentation plus dense de la forme.

4.3.1 Modèles rigides

Les distances entre les points du modèle restent constantes, seules les transformations rigides (rotations, translations . . .) s'appliquent sur le modèle. Une manière de représenter la forme du visage est de le considérer comme une surface semi-sphérique ou semi-cylindrique [La Cascia *et al.*, 2000; Sung *et al.*, 2008]. Cette hypothèse simpliste est adaptée lorsque le visage est loin de la caméra et que l'image est de faible résolution. Le faible degré de

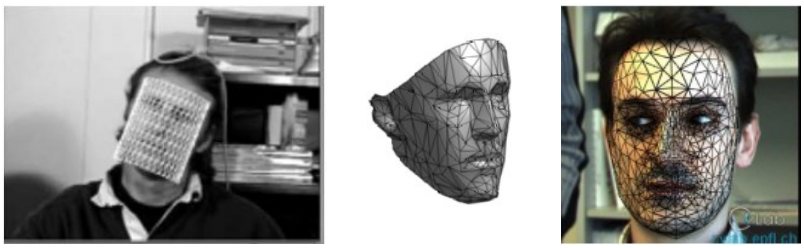


FIGURE 4.3 – Quelques exemples de modèles rigides, de gauche à droite La Cascia *et al.* [2000]; Everingham et Zisserman [2005]; Vacchetti *et al.* [2004].

liberté de ces modèles facilite le processus d'alignement puisqu'il y a peu de paramètres à optimiser. Le biais introduit par ces contraintes fortes sur le modèle peut être un handicap,

en particulier lorsque la morphologie et les expressions du visage sont trop éloignées du modèle à aligner.

4.3.2 Modèles analytiques

Les relations entre les points du modèles sont définies manuellement. Fischler et Elschlager [1992] définissent des relations élastiques (système masses-ressorts) entre les parties du modèle. Cette approche simple n'intègre pas les contraintes physiques qui sous-tendent ces déplacements. D'autres modèles tiennent compte de données anthropométriques (ouverture de la bouche et des yeux, haussement de sourcils...). Les paramètres de déformation du modèle s'appuient sur des considérations psychologiques [Pandzic et Forchheimer, 2003] et ne correspondent pas forcément aux effets du mouvement des muscles. Parke [1972] est le premier à proposer un modèle de visage capable de synthétiser des expressions en interpolant des configurations-clés du modèle. Parke développa également le premier modèle paramétrique 3D [Parke, 1974, 1982]. Depuis, un grand nombre de modèles analytiques ont vu le jour. Par exemple, le modèle Candide 3 [Ahlberg, 2001a] que nous avons déjà présenté au chapitre 2, est constitué d'un modèle rigide que l'on peut déformer à l'aide de paramètres morphologiques et d'expression. De plus, sa simplicité et son faible nombre de paramètres en font un candidat idéal pour l'analyse d'image [Dornaika et Davoine, 2008; Ahlberg, 2001b]. Xface [Balci *et al.*, 2007] est un autre exemple de modèle public, disponible en téléchargement.

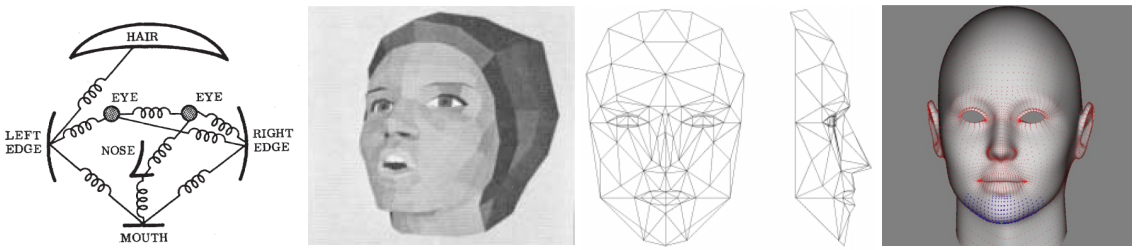


FIGURE 4.4 – Quelques exemples de modèles analytiques, de gauche à droite Fischler et Elschlager [1992]; Parke [1972]; Ahlberg [2001a]; Balci *et al.* [2007]

4.3.3 Modèles biomécaniques

Ils reproduisent les principaux éléments constituants du visage comme les os, les muscles et la peau. Cela implique une bonne connaissance de l'anatomie et une modélisation réaliste du comportement des muscles et de la peau. Waters et Terzopoulos sont pionniers dans la modélisation par muscle et proposent une série de modèles biomécaniques [Waters, 1987; Terzopoulos et Waters, 1990, 1993]. Ce type de modélisation demeure un domaine de recherche actif comme en témoigne Kähler [2007]. Le pouvoir d'expression de ces modèles est très important et le rendu est souvent plus réaliste qu'avec un modèle analytique. Ils sont

donc particulièrement appropriés au domaine de la synthèse d'image. Cette représentation n'est toutefois pas très adaptée en analyse car elle n'est pas compacte. La modification d'un paramètre (la contraction d'un muscle par exemple) a une conséquence faible et locale sur le déplacement des points du modèle. Dans ces conditions, il est difficile d'estimer les paramètres d'alignement du modèle. Ils sont tout de même utilisés dans des applications qui nécessitent un ajustement fin du modèle dans l'image. C'est le cas par exemple dans des applications de postproduction [Kalinkina *et al.*, 2007].

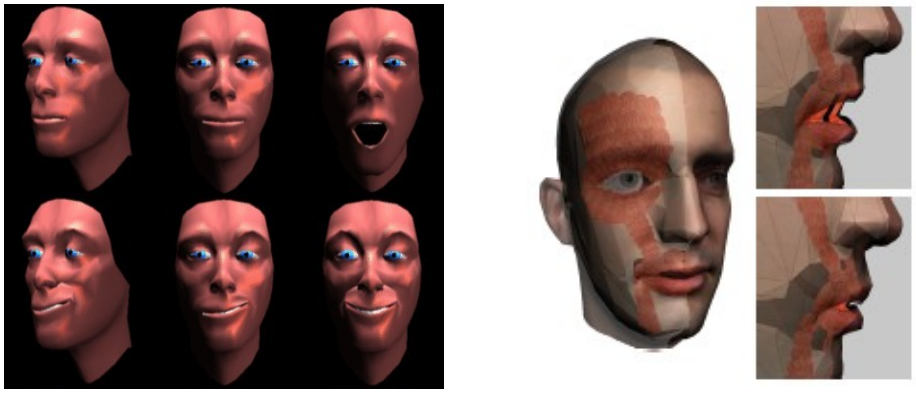


FIGURE 4.5 – Quelques exemples de modèles biomécaniques, Terzopoulos et Waters [1993]; Kähler *et al.* [2001].

Les modèles analytiques et biomécaniques sont parfois regroupés sous le terme de modèles paramétriques. Pandzic et Forchheimer [2003] proposent une taxonomie plus fine de ce type de modèles.

4.3.4 Modèles statistiques

L'objectif est de capturer les variations de forme d'un visage par l'analyse statistique d'un ensemble d'apprentissage. Plus précisément, on cherche un modèle paramétrique $\mathbf{s}(\mathbf{b})$ pour générer de nouvelles instances de formes en faisant varier le vecteur des paramètres $\mathbf{b}$. Si l'on est capable de modéliser la distribution de ces paramètres $p(\mathbf{b})$, on peut générer des formes semblables à celle de la base d'apprentissage. On peut également analyser de nouvelles formes et déterminer si elles sont plausibles.

Les modèles de forme statistiques, également appelés modèles de distribution de points (PDM, *Point Distribution Models*) sont très répandus en analyse faciale. On les retrouve notamment dans deux méthodes d'alignement populaires, les modèles actifs de forme (ASM, *Active Shape Models*) et les modèles actifs d'apparence (AAM, *Active Appearance Models*).

Modèles par ACP. Il s'agit de l'une des modélisations les plus simples et les plus utilisées. L'Analyse en Composantes Principales (ACP) capture les corrélations de déplacement

entre les points caractéristiques de la base d'apprentissage. Elle permet ainsi de décrire les principales variations d'une forme à l'aide d'un faible nombre de paramètres. La section 5.1 de ce chapitre présente en détail la création et l'utilisation de ce type de modèles.

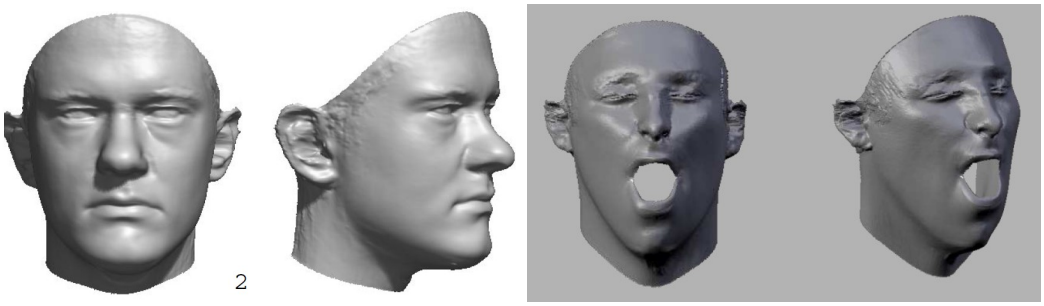
Modèles non linéaires. Les modèles par ACP reposent sur l'hypothèse que la distribution des formes est gaussienne. Ainsi, chaque instance contenue à l'intérieur d'une hyperellipsoïde dont les dimensions sont proportionnelles à la racine carrée des valeurs propres de l'ACP, est considérée comme une forme admissible par le modèle. Cependant, cette hypothèse n'est pas toujours vérifiée, lorsque la forme subit des rotations en dehors du plan image par exemple. Certaines instances du modèle ne seront donc pas plausibles, c'est-à-dire que certaines formes générées par le modèle ne ressembleront pas à une forme de visage. Le modèle doit être à la fois *complet* (qui modélise toutes les variations acceptables) et *concis* (qui ne génère pas de formes non admissibles). Romdhani *et al.* [1999] étendent la modélisation par ACP au cas non linéaire *via* une ACP par noyau. Sozou *et al.* [1995a] s'appuient sur les travaux de Kramer [1991] pour réaliser une ACP non linéaire à l'aide d'un perceptron multicouche. Sozou *et al.* [1995b] réalisent une ACP classique puis modélisent la distribution des paramètres suivant chaque axe principal par un polynôme. Ces stratégies modélisent plus fidèlement des distributions qui ne sont pas gaussiennes mais rien ne garantit que les formes générées par le modèle soient plausibles. Cootes et Taylor [1999] modélisent la distribution $p(\mathbf{b})$ par un mélange de gaussiennes dont les paramètres sont obtenus par un algorithme EM. Une forme $\mathbf{s}(\mathbf{b})$ est plausible si $p(\mathbf{b})$ est supérieur à un seuil. Ce dernier peut être choisi de manière à ce qu'un certain pourcentage (99% typiquement) d'échantillons générés aléatoirement à partir de la fonction $p(\mathbf{b})$ soit supérieur à ce seuil.

Plus récemment, Li et Ito [2005] proposent une méthode non paramétrique pour représenter la distribution des paramètres du modèle. L'idée est d'approximer cette distribution en discrétisant l'espace des paramètres. Si le modèle est défini par N_{param}, la distribution sera représentée par un tableau à N_{param} dimensions. En pratique, ce tableau est creux lorsque N_{param} est grand ou lorsqu'on dispose de peu d'exemples d'apprentissage. On s'intéresse alors à la distribution conjointe entre un paramètre et les deux paramètres qui lui sont le plus corrélés. Pour qu'une forme soit considérée comme valide, il faut que chaque paramètre de la forme se retrouve dans une case non-nulle de la table qui lui correspond. Si ce n'est pas le cas, les paramètres sont modifiés tant que cette condition n'est pas vérifiée, tout en minimisant l'impact sur le déplacement des points. Liu [2009] a montré que cette modélisation améliorait significativement la qualité de l'alignement.

4.3.5 Modèles 2D ou 3D

Le processus d'alignement est 2D par nature puisqu'il s'agit, *in fine*, de localiser un ensemble de points caractéristiques 2D dans une image 2D. Par contre, le visage que l'on souhaite décrire est tridimensionnel. Il peut donc sembler plus cohérent d'utiliser un modèle 3D. Xiao *et al.* [2004a] ont montré que la représentation d'un modèle 2D est aussi complète qu'un modèle 3D mais moins concise. En effet, un modèle 2D linéaire est capable de générer

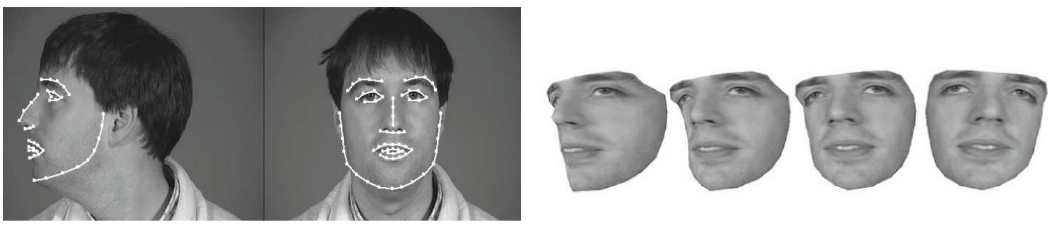
les mêmes formes qu'un modèle 3D en utilisant 6 fois plus de paramètres. La construction d'un modèle 3D statistique n'est toutefois pas aussi simple qu'un modèle 2D car l'image ne donne pas directement accès à l'information de profondeur. On peut l'estimer par triangulation sur une paire d'images stéréoscopiques [Mittrapiyanuruk *et al.*, 2004]. Blanz et Vetter [1999] utilisent un scanner laser 3D pour acquérir la structure du visage. Ces techniques donnent des modèles de bonne qualité (*cf.* figure 4.6(a)) mais elles nécessitent du matériel supplémentaire. Xiao *et al.* [2004b] et Gonzalez-Mora *et al.* [2010] contournent cette difficulté et construisent un modèle 3D à l'aide d'une méthode de *Structure from Motion*. Une



(a) (b)

FIGURE 4.6 – Modèles 3D obtenus par : (a) un scanner laser 3D [Blanz et Vetter, 1999] - (b) une méthode de *Structure from Motion* [Gonzalez-Mora *et al.*, 2010].

autre technique astucieuse consiste à construire un modèle de forme 2D pour des visages de face et d'y ajouter l'information de profondeur obtenue sur des visages de profil [Sattar *et al.*, 2007]. Le recours à des modèles 3D n'est pas pour autant une nécessité. Cootes *et al.*



(a) (b)

FIGURE 4.7 – Modèles de forme 2.5D [Sattar *et al.*, 2007]. (a) exemple d'apprentissage. (b) instances du modèle.

[2000] gèrent de grandes variations de pose du visage par un ensemble de modèles 2D. Xiao *et al.* [2004a] alignent un modèle 2D mais réduisent le pouvoir d'expression du modèle en le contraignant avec un modèle 3D.

4.4 Modèle d'apparence

Le modèle d'apparence caractérise la texture délimitée par le modèle de forme. Il existe de nombreuses manières de modéliser l'apparence. Nous les analyserons suivant 3 axes :

- La nature du modèle, génératif ou discriminant.
- La portée du modèle, global ou local.
- Les descripteurs de texture.

4.4.1 Nature du modèle

On distingue deux grandes approches de modèles. La première est dite *générative* car elle tente de modéliser l'ensemble des variations d'apparence d'une catégorie d'objet. Dans un ASM, par exemple, on modélise la distribution des niveaux de gris dans le voisinage de chaque point de la forme par une gaussienne multidimensionnelle [Cootes *et al.*, 1995]. D'autres méthodes plus élaborées telles que les AAM, permettent de synthétiser l'apparence de l'objet modélisé.

Le second type de modèle est *discriminant* car il apprend à différencier les bonnes et les mauvaises positions du modèle [Van Ginneken *et al.*, 2002; Li et Ito, 2005; Liu, 2009]. BAM (pour *Boosted Appearance Models*) illustre bien cette approche (*cf.* figure 4.8). Un classifieur à deux classes apprend *via* un algorithme de *boosting* à distinguer les bons alignements (classe positive) des mauvais alignements (classe négative). Ces approches permettent généralement de mieux délimiter les frontières d'une classe d'apparence, au détriment souvent d'un nombre d'exemples et d'un temps d'apprentissage plus important puisqu'il faut traiter à la fois des exemples positifs et négatifs. De plus, la qualité du modèle dépend beaucoup de la pertinence des exemples négatifs.

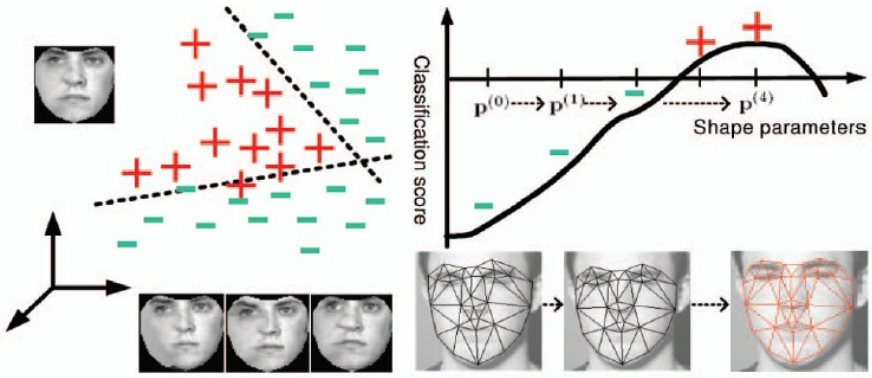


FIGURE 4.8 – Exemple d'un modèle d'apparence discriminant : le modèle apprend à distinguer les alignements corrects (+) des mauvais alignements (-). L'alignement consiste à trouver les paramètres du modèle qui maximisent le score de classification

4.4.2 Portée du Modèle

La première famille regroupe les méthodes qui caractérisent localement l'apparence du modèle. Par local, on entend une zone située autour des amers de la forme. Dans le cas des ASM, le voisinage du point est défini par le profil normal : un segment de droite centré sur l'amer et perpendiculaire au segment reliant deux points voisins (*cf.* figure 4.9). L'apparence locale $\mathbf{g}_{loc}(\mathbf{I}, \mathbf{x})$ contient les N_{pix} pixels de $\mathbf{I}$ échantillonnés le long du profil. Dans les articles récents, le voisinage des amers est souvent délimité par de petites zones rectangulaires (*patches*).

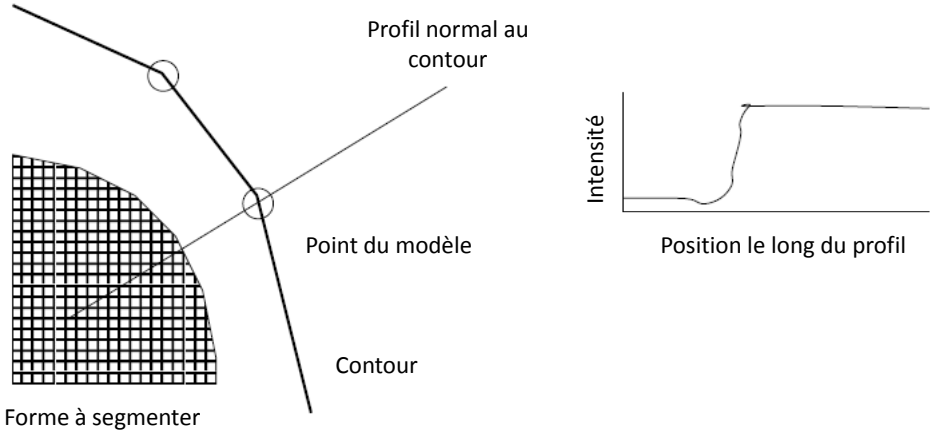
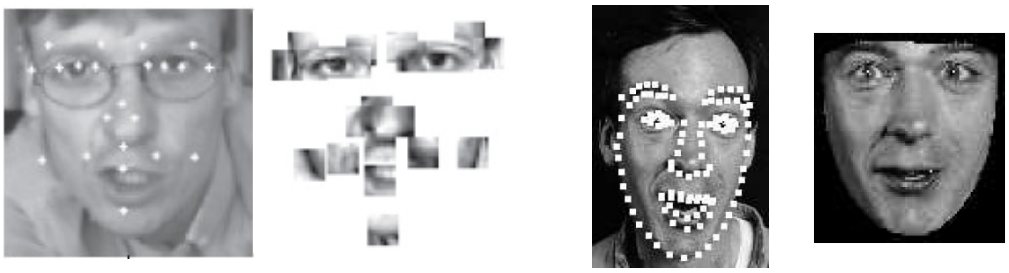


FIGURE 4.9 – Le profil normal caractérise l'apparence du voisinage d'un point du modèle [Cootes et Taylor, 2004].

Le principal inconvénient de ces méthodes est qu'elles ne tiennent pas compte de toute la texture disponible. Les approches globales de type AAM et 3DMM (*3D Morphable Models*, Blanz et Vetter [1999]) permettent de tirer profit de toute cette information. L'idée principale est de transférer la texture des exemples d'apprentissage vers un modèle de référence (la composante rigide du modèle de forme dans le cas des AAM) afin d'obtenir un ensemble de textures normalisées en forme. Chaque texture sera notée $\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s})$. La technique du transfert de texture sera détaillée dans la section 5.2.1.

4.4.3 Description de l'apparence

Qu'il soit global ou local, le modèle d'apparence correspond à une texture que l'on doit caractériser. L'approche la plus directe consiste à modéliser les variations d'intensité des pixels. La modélisation de Cootes *et al.* [1995] par une gaussienne multidimensionnelle est un exemple très simple. Cootes et Taylor [2004] ont également proposé une méthode de modélisation statistique. Elle consiste à appliquer une ACP sur la texture des visages de la base d'apprentissage. Cela n'est possible que si toutes les textures sont préalablement alignées, c'est-à-dire si elles sont transférées vers une même forme de référence pour tous les



(a) (b)

FIGURE 4.10 – Exemple d'un modèle d'apparence : (a) local [Cristinacce et Cootes, 2008] et (b) global [Cootes et Taylor, 2004].

visages. En appliquant l'ACP sur les données, on obtient un modèle linéaire d'apparence :

$$\mathbf{g}_{mod} = \bar{\mathbf{g}} + \sum_{i=1}^{N_{param}} a_i \boldsymbol{\lambda}_i \tag{4.4}$$

avec $\bar{\mathbf{g}}$ le vecteur des niveaux de gris moyens, a_i le paramètre associé au $i^{\text{ème}}$ vecteur propre $\boldsymbol{\lambda}_i$. On peut générer une nouvelle texture en modifiant ces paramètres (*cf.* figure 4.4.3).

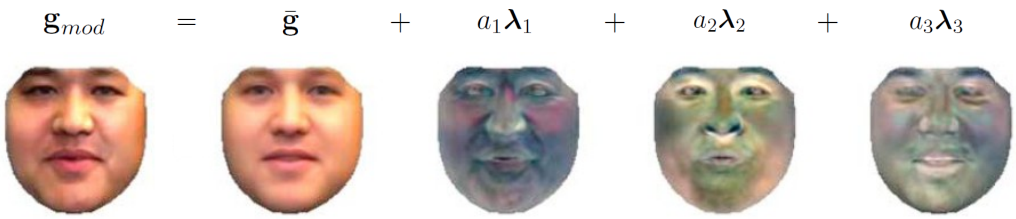


FIGURE 4.11 – Exemples d'instances du modèle d'apparence par ACP : la texture d'un visage peut s'exprimer par une texture de base (moyenne des textures de la base d'apprentissage) et une combinaison linéaire des premiers vecteurs propres de l'ACP.

On retrouve la modélisation par ACP dans des méthodes de référence telles que les AAM [Cootes et Taylor, 2004], les 3DMM, et les CLM (*Constraint Local Models*, Cristinacce et Cootes [2008]).

Modéliser directement les variations des niveaux de gris des pixels peut poser des problèmes de généralisation, en particulier lorsque les conditions d'illumination sont différentes entre la base d'apprentissage et les images de test. Pour réduire la sensibilité aux conditions d'éclairage, les représentations qui s'appuient sur les contours de l'image ont souvent été employées [Cootes et Taylor, 2001; Le Gallou, 2007; Stegmann et Larsen, 2003]. Kittipanya-ngam et Cootes [2006] ont comparé différents prétraitements des images et ont montré qu'une normalisation non-linéaire des gradients améliore à la fois la précision et la robustesse de l'alignement. Une autre manière d'améliorer la robustesse vis-à-vis de l'illu-

mination est de modéliser explicitement ces variations. Sclaroff et Isidoro [2003] incluent dans leur modèle des paramètres pour moduler le contraste et la luminance de la texture.

Une approche plus globale consiste à prendre en compte les différentes sources de variations d'apparence du visage telles que l'illumination, l'identité et les expressions faciales. Les travaux récents sur les modèles multi-linéaires [Vasilescu et Terzopoulos, 2005] offrent un moyen de les découpler. Ils ont été utilisés pour modéliser l'apparence d'un AAM [Gonzalez *et al.*, 2007; Abboud et Davoine, 2005]. Cette technique implique la constitution d'une base de données de grande dimension, représentative de l'ensemble de ces variations. Cela se traduit par une augmentation du nombre de paramètres à optimiser. Gross *et al.* [2005] montrent empiriquement qu'un modèle spécifique est plus facile à aligner qu'un modèle générique. Séguier *et al.* [2009] et Lee et Kim [2009] proposent un AAM qui s'adapte au visage à aligner. Pour Séguier *et al.* [2009] l'optimisation s'opère en deux phases. Dans un premier temps le visage est aligné avec un modèle AAM générique construit à partir d'exemples de visages inexpressifs de face. Les paramètres issus du processus d'alignement sont ensuite utilisés pour sélectionner et aligner un modèle d'apparence adapté à l'identité de la personne et qui prend en compte des variations de poses et d'expressions.

Le chapitre 3 nous a montré que la représentation par pixel était naturelle pour l'homme mais qu'elle n'était pas forcément la plus adaptée. Dans [Wiskott *et al.*, 1997], le voisinage des points du modèle est décrit par un ensemble de *jets* de Gabor. Il s'agit d'un vecteur résultant de la concaténation des amplitudes de la convolution de l'image par des filtres correspondant à des ondelettes de Gabor à différentes orientations et différentes échelles en un point de l'image (*cf.* figure 4.12). Van Ginneken *et al.* [2002] convoluent l'image

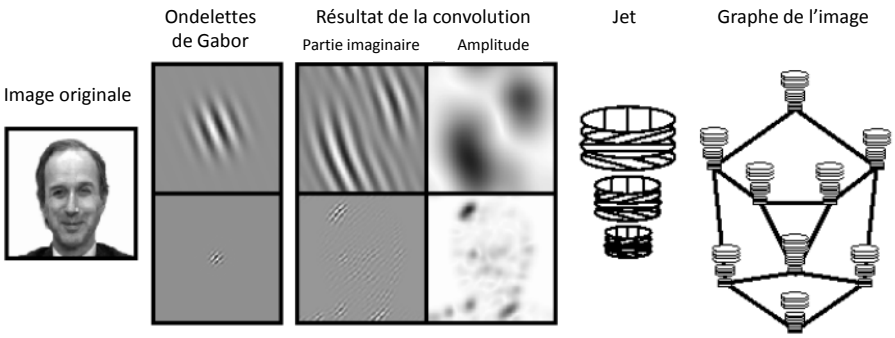


FIGURE 4.12 – Illustration d'un *jet de Gabor*.

avec des dérivées de gaussiennes à différentes échelles puis construisent des histogrammes à partir de la réponse des filtres. Les points du modèle sont caractérisés par les moments de ces histogrammes, technique standard en analyse de texture [Unser, 1986]. Il existe une multitude d'autres descripteurs pour caractériser l'apparence tels que les ondelettes de Haar [Liu, 2009; Wimmer *et al.*, 2008; Wu *et al.*, 2008].

4.5 Fonctions de coût

L'objectif de la fonction de coût est d'évaluer l'adéquation entre les paramètres du modèle et l'image. Elle s'appuie sur le modèle d'apparence décrit précédemment pour quantifier la qualité de l'alignement du modèle de forme. On cherche une fonction de coût F telle que $\underset{\mathbf{b}}{\operatorname{argmin}}\ F\left(\mathbf{I}, \mathbf{s}(\mathbf{b}), \mathbf{a}\right) = \underset{\mathbf{b}}{\operatorname{argmin}}\ d\left(\mathbf{s}(\mathbf{b}), \mathbf{s}^*\right)$.

4.5.1 Fonctions de coût empiriques

Les fonctions de coût empiriques sont le fruit d'un choix du concepteur de la méthode d'alignement. Elles découlent souvent du modèle d'apparence choisi. Ainsi, pour une modélisation de l'apparence par une distribution gaussienne, une fonction de coût potentielle sera la distance de Mahalanobis.

Dans le cas d'un ASM, on évalue la position d'un point du modèle dans l'image par :

$$f_i\left(\mathbf{I}, \mathbf{s}_i\right) = \left(\mathbf{g}_{loc}(\mathbf{I}, \mathbf{s}_i) - \bar{\mathbf{g}_i}\right)^T \mathbf{\Sigma}_i^{-1} \left(\mathbf{g}_{loc}(\mathbf{I}, \mathbf{s}_i) - \bar{\mathbf{g}_i}\right) \tag{4.5}$$

avec $\mathbf{g}_{loc}(\mathbf{I}, \mathbf{s}_i)$ le vecteur des pixels du profil normal du $i^{\text{ème}}$ point du modèle. La matrice de covariance $\mathbf{\Sigma}_i$ et la moyenne des niveaux de gris $\bar{\mathbf{g}_i}$ de ce point ont été estimées sur un ensemble d'apprentissage et constituent le modèle d'apparence. La fonction de coût locale $f_i(\mathbf{I}, \mathbf{s}_i)$ est proportionnelle au log de la probabilité que l'observation $\mathbf{g}_{loc}(\mathbf{I}, \mathbf{s}_i)$ soit issue de la distribution. La fonction de coût de l'ASM correspond à la somme des distances pour les N_{pt} points du modèle de forme :

$$F(\mathbf{I}, \mathbf{b}) = \sum_{i=1}^{N_{pt}} f_i(\mathbf{I}, \mathbf{s}_i(\mathbf{b})) \tag{4.6}$$

Certains modèles sont capables de synthétiser de grandes variétés d'apparences de visages (AAM, 3DMM). L'objectif est de générer l'instance du modèle qui ressemble le plus à l'image observée. Cette ressemblance est quantifiée par l'erreur quadratique moyenne entre $\mathbf{g}_{mod}$ le vecteur de pixels du modèle et $\mathbf{g}_{glob}$ celui de l'observation [Cootes et Taylor, 2004].

$$F(\mathbf{I}, \mathbf{b}, \mathbf{a}) = \left\| \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b})) - \mathbf{g}_{mod}(\mathbf{a}) \right\|^2 \tag{4.7}$$

Cette fonction de coût présente deux inconvénients. D'une part, le pouvoir d'expression du modèle doit être suffisamment important pour synthétiser l'apparence du visage dans l'image observée. Sinon, le score de la fonction de coût ne sera pas nécessairement minimum pour le bon jeu de paramètres du modèle. D'autre part, cette fonction de coût peut contenir des minima locaux, ce qui restreint le rayon de convergence du modèle.

Romdhani et Vetter [2005] apportent une réponse à ce problème en combinant linéairement plusieurs modèles d'apparence afin de lisser la fonction de coût. L'idée directrice est

que les fonctions de coût associées à chaque modèle d'apparence contiennent des minima locaux mais que la position de ces minima dans l'espace des paramètres varie en fonction du modèle d'apparence ; à l'exception du minimum global qui correspond au bon jeu de paramètres. Ainsi la combinaison des différents modèles aura pour effet de lisser la fonction de coût par compensation des minima locaux et de renforcer le minimum global. La figure 4.13 illustre ce principe. La première ligne montre différentes caractéristiques issues de l'image (*image features*). Des connaissances *a priori* sur le modèle (*model features*) de forme et de texture sont également introduites. La deuxième ligne représente les fonctions de coût de chaque apparence en fonction des rotations du modèle suivant l'angle pan. Leur combinaison améliore la fonction de coût qui sera minimisée (*fitting*). Les performances de cette solution restent toutefois très dépendantes du concepteur puisque le choix des descripteurs images et du poids de leur contribution dans la fonction de coût est empirique.

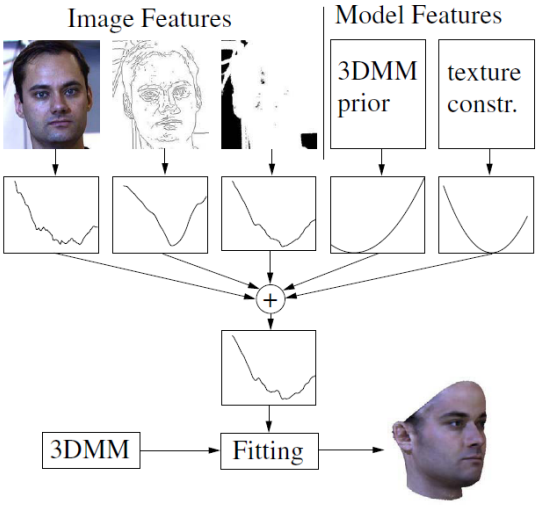


FIGURE 4.13 – Une combinaison linéaire de différents modèles d'apparence améliore l'aspect de la fonction de coût [Romdhani et Vetter, 2005]

Gacon *et al.* [2005] proposent une fonction originale permettant de prendre en compte des variations de l'apparence fortement non-linéaire. Ils utilisent les filtres de dérivées de gaussiennes pour caractériser localement l'apparence. L'idée est de prédire, à l'aide d'un réseau de neurones, la réponse de ces filtres en fonction des paramètres de forme du modèle. La fonction de coût est alors la différence entre la prédiction du réseau et la valeur des descripteurs.

4.5.2 Fonctions de coût adaptées

Nous venons de voir que les fonctions de coût empiriques présentaient deux inconvénients majeurs. Lorsqu'on perturbe les paramètres du modèle de forme, la fonction de coût peut présenter des minima locaux. Cette particularité est très gênante pour le processus d'optimisation car il peut converger vers cette solution. Les approches locales d'optimisation

telles que la descente de gradient sont particulièrement sensibles à ces minima locaux. Le second inconvénient est que l'un de ces minima peut correspondre au minimum global de la fonction. Dans ce cas, même l'algorithme d'optimisation le plus performant ne pourrait pas trouver les bons paramètres du modèle. Ces problèmes apparaissent principalement parce que le modèle d'apparence et la fonction de coût sont conçus sans considérer l'information au voisinage des bons paramètres du modèle. Cette idée, illustrée dans la figure 4.14, consiste donc à apprendre une fonction de coût adaptée en utilisant durant la phase d'apprentissage à la fois l'apparence des visages $\mathbf{g}(\mathbf{I}, \mathbf{s}(\mathbf{b}_\mathbf{I}^*))$ avec les bons paramètres du modèle $\mathbf{b}_\mathbf{I}^*$ et l'apparence au voisinage $\mathbf{g}(\mathbf{I}, \mathbf{s}(\mathbf{b}_\mathbf{I}^* + \Delta\mathbf{b}))$ (figure 4.14b).

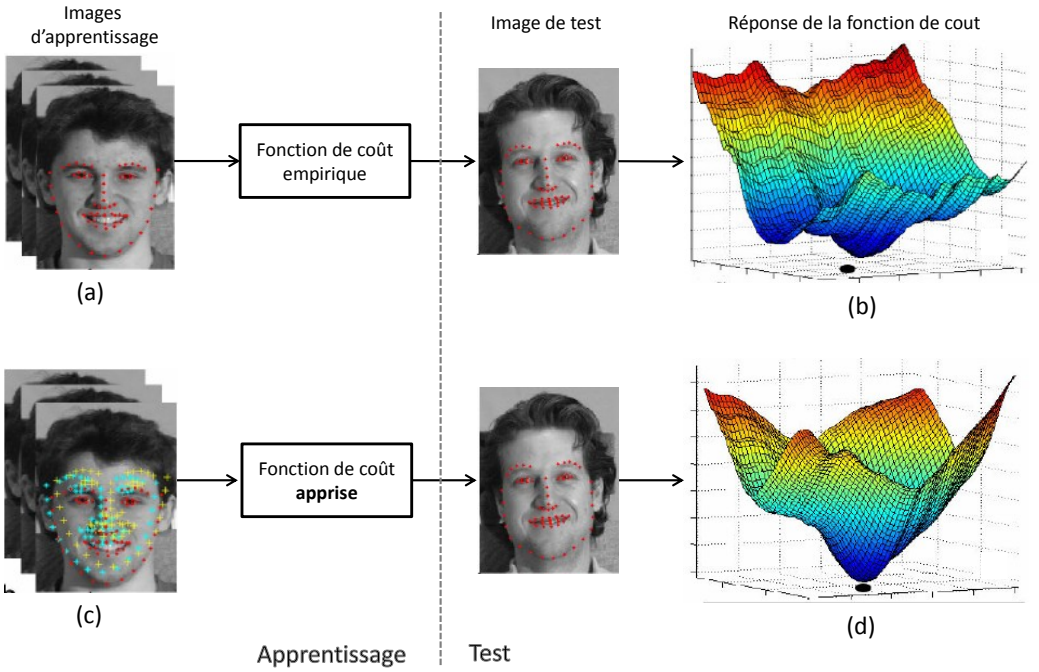


FIGURE 4.14 – Principe d'une fonction de coût adaptée. (a) Seule l'apparence des visages bien alignés est exploitée. (b) La fonction de coût a plusieurs minima locaux et son minimum global ne correspond pas aux bons paramètres du modèle (point noir). (c) La conception de la fonction de coût prend en compte l'apparence du visage au voisinage des bons paramètres du modèle. (d) La surface de la fonction de coût apprise est améliorée et le minimum global est à la bonne position. (Schéma inspiré de Nguyen et De la Torre, 2010).

4.5.2.1 Approches par classification

Elles s'appuient sur un modèle d'apparence discriminant (*cf.* section 4.4.1). L'objectif est de construire un classifieur capable d'identifier les bonnes positions du modèle. Li et Ito [2005] utilisent l'algorithme AdaBoost pour apprendre les bonnes positions des points du modèle. Ils montrent expérimentalement qu'une fonction de coût locale, définie par la sortie

du classifieur, donne de meilleurs résultats que la distance de Mahalanobis. La figure 4.15 montre un exemple des réponses de la fonction de coût le long du profil normal.

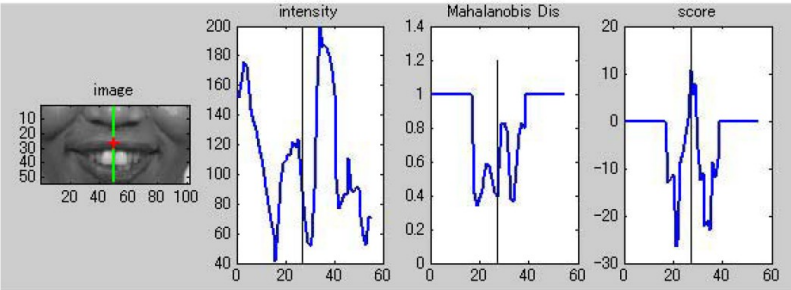


FIGURE 4.15 – Image de gauche : un point du modèle (point rouge) et son profil normal (ligne verte). Graphique 1 : distribution des niveaux de gris le long du profil. Graphique 2 : distance de Mahalanobis le long du profil. Graphique 3 : score par AdaBoost le long du profil

Les *Boosted Appearance Models* (BAM) de Liu [2009] exploitent une idée similaire. L'algorithme *GentleBoost* est utilisé pour sélectionner les filtres de Haar pertinents pour distinguer les bons alignements des mauvais. Les descripteurs de Haar sont calculés sur un modèle d'apparence global.

Van Ginneken *et al.* [2002] proposent une autre manière d'utiliser un classifieur pour évaluer la qualité de l'alignement. Un classifieur H est entraîné à répondre 1 si un pixel appartient à la forme à segmenter et 0 sinon. La fonction de coût locale est la somme des différences absolues entre la sortie du classifieur et la sortie attendue. Pour chaque élément $\mathbf{g}_{loc}^{(j)}$ du profil normal :

$$f_i(\mathbf{I}, \mathbf{s}_i) = \sum_{j=-k}^{1} H(\mathbf{g}_{loc}^{(j)}(\mathbf{I}, \mathbf{s}_i)) + \sum_{j=0}^{k} 1 - H(\mathbf{g}_{loc}^{(j)}(\mathbf{I}, \mathbf{s}_i)) \tag{4.8}$$

avec les indices le long du profil $\mathbf{g}_{loc}$ allant de $-k$ à k (profil orienté de l'extérieur vers l'intérieur de l'objet).

Les approches par classification donnent de bons résultats car la position du minimum de la fonction de coût est apprise en tenant compte de l'apparence au voisinage de la bonne position du modèle. Toutefois, utiliser la sortie du classifieur comme fonction de coût ne nous garantit pas que le coût augmente à mesure que l'on s'éloigne de la bonne position.

4.5.2.2 Approches par régression

L'idée directrice, illustrée dans la figure 4.16, est de définir une fonction de coût idéale (figure 4.16b) et d'apprendre, à partir d'exemples plus ou moins bien alignés (figure 4.16d), une fonction avec un comportement similaire (figure 4.16e). Ces approches ne considèrent pas l'alignement comme un problème binaire (bien/mal aligné) mais comme un processus

continu, et cherchent à répondre au problème des minima locaux qui ne sont pas pris en compte dans les approches par classification.

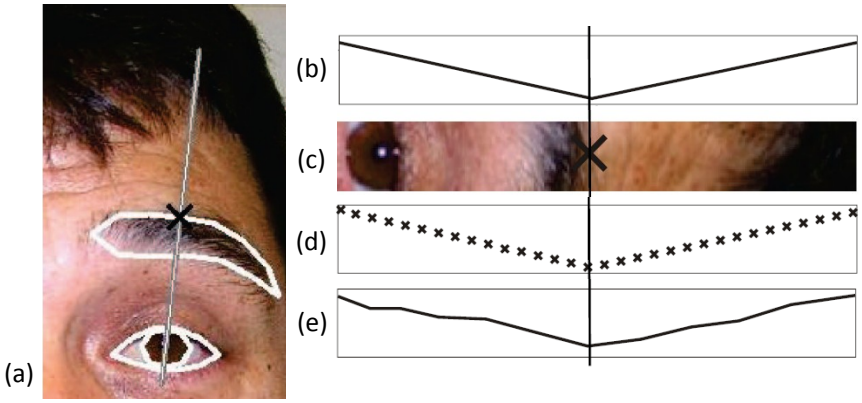


FIGURE 4.16 – (a) Profil d'un point du sourcil. (b) Fonction de coût idéale. (c) Image le long du profil. (d) Données d'apprentissage. (e) Fonction de coût apprise [Wimmer *et al.*, 2008]

Fonctions de coût idéales Intuitivement, on souhaite que la valeur retournée par la fonction de coût :

- Soit minimum lorsque le modèle est bien aligné.
- Augmente à mesure que le modèle s'éloigne de sa bonne position.

Ces deux caractéristiques définissent la fonction de coût idéale. Wimmer *et al.* [2008] ont formalisé les deux propriétés d'une telle fonction de coût :

- La propriété d'exactitude **P1** : le minimum global de la fonction de coût correspond à la meilleure position du modèle. La propriété est vérifiée si pour tout $\mathbf{b}$ tel que $\mathbf{b} \neq \mathbf{b}_{\mathbf{I}}^{*}$ alors :

$$F\left(\mathbf{I}, \mathbf{b}_{\mathbf{I}}^{*}\right) < F\left(\mathbf{I}, \mathbf{b}\right) \tag{4.9}$$

- La propriété d'unimodalité **P2** : la fonction de coût n'a ni extrema locaux ni points selles. La propriété est vérifiée s'il existe un $\mathbf{m}$ unique qui pour tout $\mathbf{b} \neq \mathbf{m}$:

$$F\left(\mathbf{I}, \mathbf{m}\right) < F\left(\mathbf{I}, \mathbf{b}\right) \tag{4.10}$$

et si pour tout $\mathbf{b} \neq \mathbf{m}$ alors :

$$\nabla F\left(\mathbf{I}, \mathbf{b}\right) \neq 0; \tag{4.11}$$

La figure 4.17 montre quatre exemples de fonctions avec et sans les propriétés **P1** et **P2**. La ligne en pointillés correspond au meilleur jeu de paramètres $\mathbf{b}_{\mathbf{I}}^{*}$. Un algorithme d'optimisation, qui s'appuie sur la fonction de coût avec ces deux propriétés, est sûr que le minimum local trouvé corresponde au minimum global de la fonction, et que la solution coïncide avec le meilleur jeu de paramètres du modèle.

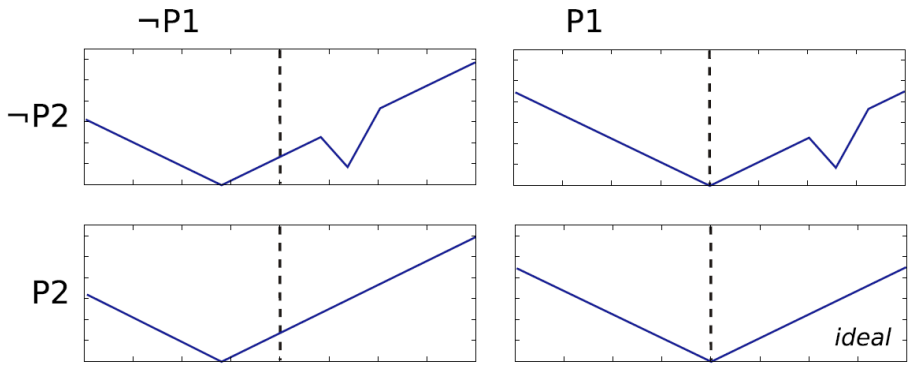
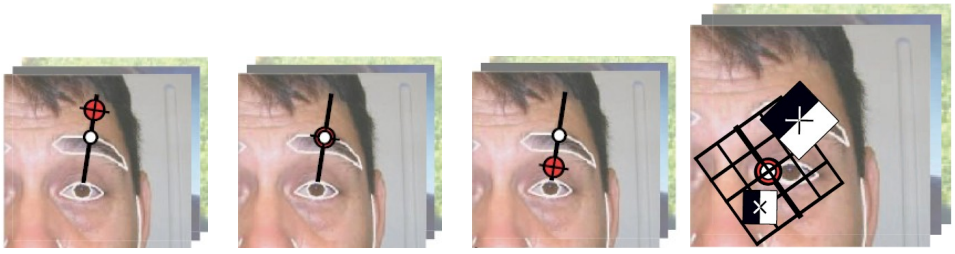


FIGURE 4.17 – Chaque figure correspond à un exemple de fonction de coût satisfaisant ou non les propriétés **P1** et **P2**. La fonction de coût est dite idéale lorsque **P1** et **P2** sont respectées. Illustration tirée de Wimmer *et al.* [2008]

Apprentissage de la fonction de coût Wimmer *et al.* [2008] proposent une instance de la fonction de coût. Il s'agit d'une fonction de coût locale qui mesure la distance entre un point dans l'image $\mathbf{x}$ et un point de la forme bien aligné $\mathbf{s}_i(\mathbf{b}^*_\mathbf{I})$:

$$f^*_i(\mathbf{I}, \mathbf{x}) = |\mathbf{x} - \mathbf{s}_i(\mathbf{b}^*_\mathbf{I})| \tag{4.12}$$

A partir de cette définition et d'une base de données annotée, Wimmer et al. génèrent automatiquement des exemples d'apprentissage en déplacant chaque point du modèle le long de son profil normal (*cf.* figure 4.18(b)). Chaque point est décrit par un ensemble d'ondelettes de Haar disposées sur une grille qui est centrée sur le point et orientée suivant son profil (figure 4.18(b)). La fonction de coût idéale (équation (4.12)) est apprise par un arbre de régression qui réalise simultanément le processus d'apprentissage et de sélection des ondelettes.



(a) (b)

FIGURE 4.18 – (a) Les exemples d'apprentissage sont choisis uniformément le long du profil. (b) Un point est caractérisé par un ensemble d'ondelettes de Haar sélectionnées automatiquement.

Brunet *et al.* [2009] proposent une autre façon d'apprendre la fonction de coût idéale. Ils partent du constat que l'erreur de reconstruction (définie précédemment par l'équation (4.7)) n'est pas une fonction de coût convexe pour les modèles d'apparence par ACP. Ils

définissent la réponse de la fonction de coût désirée f^* par une gaussienne inversée, dont le minimum correspond à la position correcte du point dans l'image et les axes correspondent aux directions de déplacement dans l'image. Ces déplacements $\Delta\mathbf{x}$ s'effectuent dans le voisinage $\mathcal{N}$ de la bonne position du point dans l'image. Brunet *et al.* cherchent alors un espace de projection $\mathbf{B}$ dans lequel l'erreur de reconstruction se comporte comme la fonction de coût désirée. Ce sous-espace est obtenu en minimisant :

$$E_i = \sum_{j=1}^{N_{im}} \sum_{\Delta\mathbf{x}\in\mathcal{N}} \left(\alpha_j f^*(\Delta\mathbf{x}) + \beta_j - \frac{\|\mathbf{g}_{loc}(\mathbf{I}_j, \mathbf{s}_i + \Delta\mathbf{x}) - \bar{\mathbf{g}}_{mod} - \mathbf{B}\mathbf{a}_j\|^2}{\|\mathbf{g}_{loc}(\mathbf{I}_j, \mathbf{s}_i + \Delta\mathbf{x})\|^2} \right)^2 \tag{4.13}$$

par rapport à $\mathbf{B}$, $\mathbf{a}_j$, α_j et β_j, pour les N_{im} de la base d'apprentissage. Les coefficients α_i et β_i sont des facteurs d'échelle et de décalage de la fonction de coût pour chaque image. Brunet et al. proposent une méthode itérative pour minimiser cette grandeur : ils cherchent alternativement une solution analytique pour α_i et β_i puis une solution pour $\mathbf{B}$ par une descente de gradient.

Nguyen et De la Torre [2010] définissent une fonction de coût quadratique qui peut s'appliquer à de nombreux modèles d'apparence (ASM, AAM, CLM ...). Cette fonction est de la forme :

$$F(\mathbf{I}, \mathbf{b}) = \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}))^T \mathbf{A} \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b})) + 2\mathbf{c}^T \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b})) \tag{4.14}$$

Apprendre la fonction de coût revient à estimer les valeurs de la matrice symétrique $\mathbf{A}$ et du vecteur $\mathbf{c}$ en respectant deux contraintes :

- Un minimum local à la bonne position : le gradient de la fonction de coût est nul pour $\mathbf{b} = \mathbf{b}^*_\mathbf{I}$.
- Pas d'autres minima dans le voisinage de $\mathbf{b}^*_\mathbf{I}$: la direction indiquée par la descente de gradient et le vecteur de déplacement vers la bonne position $\mathbf{b}^*_\mathbf{I}$ doivent être colinéaires.

Nguyen et De la Torre [2010] montrent que si l'on utilise un développement en série de Taylor du premier ordre pour approximer le gradient de la fonction de coût, l'apprentissage se ramène à un problème d'optimisation quadratique avec des contraintes linéaires sur $\mathbf{A}$ et $\mathbf{c}$.

Les *Boosted Ranking Models* (BRM), proposés par Wu *et al.* [2008], conservent le modèle de forme et d'apparence des BAM mais utilisent une manière originale d'apprendre une fonction de coût convexe dans le voisinage du bon alignement. Wu et al. restreignent le problème d'apprentissage de cette fonction à la conception d'un classifieur qui détermine si une modification des paramètres du modèle de forme améliore l'alignement. Un classifieur fort apprend par *boostingbrunet09iccvw* laquelle des deux observations, $\mathbf{g}_{glob}(\mathbf{I}, \mathbf{b}_1)$ et $\mathbf{g}_{glob}(\mathbf{I}, \mathbf{b}_2)$, correspond au meilleur alignement. Pour y parvenir, il faut constituer un ensemble d'apprentissage contenant des exemples négatifs (mal alignés) et positifs (mieux alignés). Pour chaque image labélisée, on tire aléatoirement $N_{\Delta b}$ vecteurs de perturbation $\Delta\mathbf{b}_u$ qui serviront à créer l'ensemble d'apprentissage (*cf.* figure 4.19(a)). Les exemples positifs sont définis par des paires ordonnées de textures normalisées en forme (transférées vers une forme de référence)

$$\Big(\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}_i + v\Delta\mathbf{b}_u)), \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}_i + (v+1)\Delta\mathbf{b}_u))\Big) \tag{4.15}$$

avec v un entier positif qui module l'amplitude de la déformation. A chaque exemple positif correspond un exemple négatif $\Big(\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}_i + (v+1)\Delta\mathbf{b}_u)), \mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}_i + v\Delta\mathbf{b}_u))\Big)$. Ils utilisent ensuite l'algorithme GentleBoost pour entrainer des classifieurs faibles. Ces derniers sont des filtres de Haar dont la réponse est seuillée. A la sortie de l'algorithme, on obtient une fonction de coût définie par la somme des classifieurs faibles sélectionnés (*cf.* figure 4.19(b)).

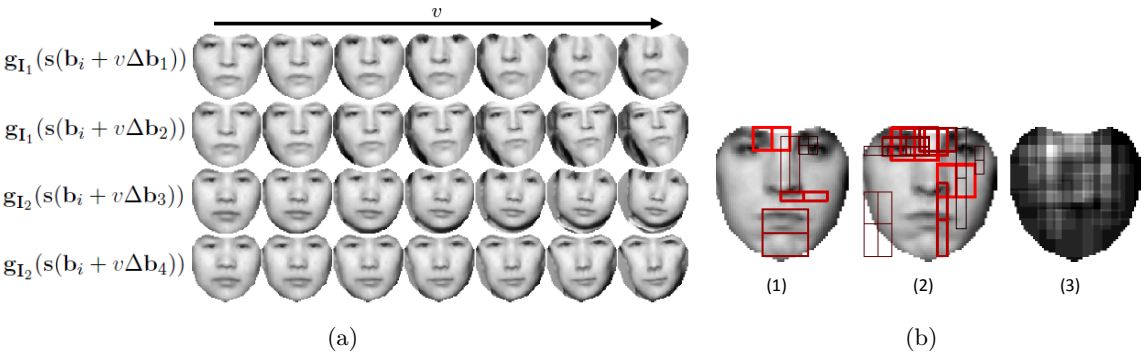


FIGURE 4.19 – *Boosted Ranking Models* [Wu *et al.*, 2008]. (a) Exemples d'apprentissage (b) Filtres de Haar sélectionnés : (1) les 5 premiers (2) les 10 suivants (3) la densité spatiale des 50 premiers filtres.

4.6 Méthodes de recherche

L'étape de recherche consiste à trouver les paramètres qui ajustent au mieux le modèle dans l'image. Cette section présente un tour d'horizon des principales méthodes d'estimation de paramètres pour l'alignement. On distingue les recherches globales, qui recherchent une solution dans tout l'espace des paramètres (grid search ou méthodes par vote par exemple), des recherches locales qui visent à améliorer séquentiellement une estimation grossière de la solution (descente de gradient, optimisation de Gauss-Newton...).

4.6.1 Recherche globale

Une recherche exhaustive ou naïve (*grid search*, recherche purement aléatoire...) n'est pas envisageable car la solution $\tilde{\mathbf{b}}$ est dans $\mathbb{R}^{N_{param}}$, avec N_{param} le nombre de paramètres du modèle pouvant être relativement grand. D'autres méthodes déterministes, telles que la transformée de Hough généralisée [Ballard, 1987], peuvent être utilisées. Elles reposent sur une discrétisation de l'espace des paramètres qui implique un compromis entre temps de calcul et précision. Les méthodes stochastiques sont plus répandues pour aligner un visage. Sattar *et al.* [2008] utilisent un algorithme génétique pour optimiser les paramètres d'un modèle 2.5D. Kalinkina [2009] aligne finement certaines zones de son modèle à l'aide d'un recuit simulé. Des filtres particulaires sont souvent mis en œuvre dans des applications de

suivi [Dornaika et Davoine, 2008] mais ils sont difficilement exploitables lorsque la zone de recherche ou la dimension de l'espace est trop grande. Dans l'ensemble, les méthodes d'optimisation globale ont une bonne couverture de l'espace de recherche au détriment de la charge de calcul. Elles seront donc bien adaptées lorsqu'on n'a pas d'estimation grossière de la solution, lorsque la fonction de coût à optimiser a de nombreux mimina locaux ou lorsque le nombre de paramètres à optimiser est faible. On pourra également utiliser le résultat d'une optimisation globale comme initialisation d'une optimisation locale.

4.6.2 Recherche locale

Les méthodes locales partent d'une estimation grossière de la solution et cherchent à l'améliorer, le plus souvent par un processus itératif. Ces méthodes sont donc très sensibles aux minima locaux et sont bien adaptées aux fonctions de coût convexes. On distingue les méthodes orientées projections de celles orientées paramètres. La première catégorie cherche dans l'image la meilleure position de chaque point indépendamment des positions relatives des autres points du modèle, puis cherche une solution admissible des paramètres du modèle qui décrit au mieux l'agencement des points dans l'image. La seconde catégorie recherche la meilleure position du modèle en agissant directement sur les paramètres du modèle.

4.6.2.1 Recherche orientée projections

On peut décomposer le processus d'estimation des paramètres du modèle en trois étapes. Dans un premier temps, le modèle $\mathbf{s}$ est projeté dans l'image à partir d'une première estimation de $\mathbf{b}$. La position de chaque composante du modèle $\mathbf{s}_i$ est ensuite optimisée séparément. Il s'agit de trouver dans l'image une position voisine de $\mathbf{s}_i$ qui minimise la fonction de coût locale f_i. Il est intéressant de noter que la recherche s'effectue dans l'image et donc dans un espace à 2 dimensions. Enfin, on estime les paramètres du modèle qui décrivent au mieux les nouvelles positions des points du modèle dans l'image. Ces trois étapes sont souvent réalisées itérativement. L'ASM est le représentant le plus populaire de cette catégorie d'alignement.

L'espace de recherche à deux dimensions constitue le principal avantage des approches par projections puisqu'il permet une recherche exhaustive de la bonne position pour chaque point du modèle. Ces méthodes présentent toutefois d'importantes limitations :

1. Elles nécessitent une fonction de coût par point. Cet aspect peut être contraignant lorsque le nombre de points du modèle est important et que la fonction de coût est apprise.
2. Si les fonctions de coût locales sont convexes, rien ne garantit que la fonction de coût globale le soit également.
3. Ce type d'optimisation est sensible aux données aberrantes et peut facilement diverger. En effet, les contraintes sur le modèle étant relâchées dans la deuxième phase

de l'algorithme, un point du modèle peut s'éloigner de sa bonne position et attirer le modèle vers lui.

4.6.2.2 Recherche orientée paramètres

Ce type de méthode agit directement sur le vecteur de paramètres pour trouver le modèle qui décrit au mieux le contenu de l'image. Pour trouver la bonne solution, on peut soit estimer la modification des paramètres à effectuer pour améliorer l'alignement, soit optimiser une fonction de coût.

Apprentissage du déplacement On apprend une fonction de correspondance entre l'apparence $\mathbf{g}(\mathbf{I}, \mathbf{s}(\mathbf{b}))$ et la mise à jour des paramètres $\Delta\mathbf{b}$ à partir d'un ensemble d'apprentissage :

$$\{\mathbf{g}(\mathbf{I}_j, \mathbf{s}(\mathbf{b}_\mathbf{I}^* + \Delta\mathbf{b}_i)), \Delta\mathbf{b}_i\}_{i=1\ldots N_{def}, j=1\ldots N_{im}} \tag{4.16}$$

avec N_{def} le nombre de perturbations et N_{im} le nombre d'image de la base d'apprentissage ; Dans la formulation des AAM de Cootes *et al.* [1998], la mise à jour des paramètres de forme et d'apparence est apprise par une régression linéaire multivariée.

$$\mathbf{b} \leftarrow \mathbf{b} - \mathbf{A}(\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b})) - \mathbf{g}_\mathbf{mod}) \tag{4.17}$$

Si la solution est loin de l'approximation initiale, les premières estimations de $\mathbf{b}$ ne seront pas précises mais l'erreur entre l'apparence observée $\mathbf{g}_{glob}(\mathbf{I}, \mathbf{s}(\mathbf{b}))$ et l'apparence du modèle $\mathbf{g}_{mod}$ devrait diminuer. Le processus devrait alors converger après plusieurs itérations. Différentes solutions ont été proposées pour améliorer l'apprentissage de la fonction de correspondance. Ybanez *et al.* [2007] ou Donner *et al.* [2006] utilisent par exemple une Analyse Canonique des Correspondances (ACC). Les *displacement experts* de Williams *et al.* [2005] s'appuient sur un RVM Tipping [2001]. Ces méthodes n'ont théoriquement pas besoin de fonction de coût puisqu'elles intègrent un critère d'arrêt : lorsque le vecteur de déplacement est nul ou après quelques oscillations. Elles sont toutefois souvent couplées avec une fonction de coût (l'erreur de reconstruction dans le cas de AAM) ou un détecteur [Williams *et al.*, 2005; Li *et al.*, 2007a] pour valider l'alignement.

Optimisation de la fonction de coût L'alignement consiste à minimiser la fonction de coût

$$\tilde{\mathbf{b}} = \underset{\mathbf{b}}{\operatorname{argmin}}\ F(\mathbf{I}, \mathbf{b}, \mathbf{a}) \tag{4.18}$$

Les approches par descente de gradient ont naturellement été explorées pour optimiser cette fonction. Blanz et Vetter [1999] utilisent une méthode de gradient stochastique qui évite en partie les minima locaux en ajoutant du bruit dans l'estimation du gradient. Le problème principal des méthodes par descente de gradient concerne l'estimation du pas. Sclaroff et Isidoro [2003] utilisent une méthode du second ordre, l'algorithme de Levenberg-Marquardt pour estimer ce pas. Cette méthode repose cependant sur l'estimation de la matrice Hessienne qui est gourmande en temps de calcul et en place mémoire.

Pour les AAM, la fonction de coût la plus utilisée est l'erreur quadratique de reconstruction donnée par l'équation (4.7). L'algorithme d'optimisation de Gauss-Newton est particulièrement bien adapté pour ce type de problème. Il nécessite toutefois de connaître ou d'estimer la matrice Jacobienne de F par rapport à $\mathbf{b}$. Cootes *et al.* [1998] font l'hypothèse que cette matrice est constante et que, par conséquent, elle peut être pré-calculée. La méthode *projected out* de Matthews et Baker [2004], une adaptation aux AAM de la méthode de composition inverse [Matthews et Baker, 2004], a permis de relâcher la contrainte sur la Jacobienne. C'est une méthode très rapide mais son rayon de convergence et son pouvoir de généralisation sont relativement faibles [Gross *et al.*, 2005].

4.6.2.3 Méthodes heuristiques

La sensibilité aux minima locaux constitue l'inconvénient majeur des méthodes locales d'optimisation. De nombreuses stratégies ont été déployées pour les éviter. Cootes et Taylor [2001] alignent le modèle sur une image dont la résolution augmente à mesure que l'on se rapproche de la bonne solution. Ceci a pour effet de lisser la fonction de coût dans les premières itérations. Dornaika et Ahlberg [2006] proposent une forme de relaxation en approximant successivement les paramètres de transformation rigide et les paramètres de déformation du modèle. Mercier [2007] obtient de meilleurs alignements en découplant ces deux paramètres. Il propose également l'utilisation d'une fonction de coût robuste pour améliorer alignement en cas d'occultation du visage.

4.7 Discussion

Cet état de l'art a mis en évidence l'importance de la fonction de coût dans le processus de l'alignement de visages. Récemment, le choix et la conception de cette fonction ont reçu une attention accrue. La méthode que nous présentons au chapitre suivant s'inscrit dans cette lignée. Nous avons privilégié une approche par régression car elle permet d'apprendre une fonction de coût avec peu de minima locaux et un minimum global qui correspond au meilleur ajustement du modèle dans l'image. Encouragés par les résultats obtenus pour l'estimation de la pose de la tête, nous avons choisi d'utiliser BISAR pour approximer la fonction de coût idéale. Ce sera églament l'occasion de valider BISAR sur une nouvelle problématique.

On souhaite laisser le choix à l'algorithme des zones de textures les plus pertinentes pour évaluer l'alignement. Nous avons donc naturellement privilégié un modèle d'apparence globale qui ne fait pas d'hypothèse *a priori* sur les régions pertinentes. Par ailleurs, l'utilisation d'une fonction de transfert de la texture (*warping*) vers une forme constante permet de rendre l'apparence invariante aux changements de pose, d'expression et de morphologie du visage. Nous avons également conservé les mêmes descripteurs de texture que pour l'estimation de la pose. Les filtres de Haar ont d'ailleurs été utilisés avec succès dans les méthodes d'alignement [Wu *et al.*, 2008; Wimmer *et al.*, 2008].

Le modèle de forme est standard. Nous avons opté pour un modèle statistique 2D par ACP pour :

- Son adaptabilité aux données : contrairement aux modèles analytiques tel que Candide, il est très facile d'adapter parfaitement un modèle dans une image lorsque la position des points caractéristiques dans l'image est connue. Cet aspect est d'autant plus important que nous avons besoin dans notre méthode d'exemples de modèles plus ou moins bien alignés.
- Sa diffusion : c'est un modèle très répandu et éprouvé en alignement de visage.
- Sa compacité : les modèles statistiques permettent de modéliser de nombreuses variations avec un faible nombre de paramètres.
- Sa facilité de mise œuvre : ils ne nécessitent qu'une base de données d'images annotées.

La difficulté de notre approche repose sur l'apprentissage de la fonction de coût. Si cette étape s'est bien passée, c'est-à-dire si la fonction apprise est relativement convexe dans le voisinage de la bonne position, alors une méthode simple d'optimisation locale telle qu'une descente de gradient peut être envisagée. Nous utiliserons la méthode proposée par Dornaika et Ahlberg [2006] car elle est simple à implémenter, a peu de paramètres à ajuster et ne nécessite pas de matrices jacobiennes empiriques ou analytiques.

Chapitre 5

Alignement par apprentissage de la fonction de coût

Dans ce chapitre, nous présentons BiBAM (pour *Bisar Based Appearance Model*), notre nouvelle méthode d'alignement d'un modèle déformable dans une image.

5.1 Modèle de forme

Le modèle de forme est le modèle de distribution de points (PDM de l'anglais *Point Distribution Model*) couramment utilisé dans les ASM et les AAM. Il s'agit d'un modèle filaire défini par la position d'un ensemble de points 2D. Mathématiquement, un modèle de forme $\mathbf{s}$ est défini par les coordonnées des N_{pt} points qui le composent :

$$\mathbf{s} = (x_1, y_1, x_2, y_2, \ldots, x_{N_{pt}}, y_{N_{pt}})\,; \tag{5.1}$$

L'objectif du PDM est de modéliser les variations de forme propres au visage à partir d'un ensemble d'exemples. Chaque exemple correspond à un agencement spécifique des points du modèle issus d'une image de visage labélisée. La figure 5.1 présente un exemple de visage annoté avec 37 points caractéristiques.

Nous distinguerons deux types de paramètres :

– Les paramètres externes : ils modélisent les transformations géométriques liées à la position et à la focale de la caméra ainsi que la position, l'orientation et la taille du visage dans le plan image.

– Les paramètres internes : ils correspondent à toutes les autres sources de variations. Il s'agit principalement des variations de morphologie et d'expression du visage ; en 2D ces paramètres modélisent également les rotations hors-plan de la tête.

Comme au chapitre 4 nous appellerons déformation les variations des paramètres internes. Ces déformations seront modélisées par une analyse statistique. Par contre les paramètres externes ont une formulation analytique simple et compacte. Nous allons donc, dans un premier temps, supprimer les variations attribuables à ces transformations externes.

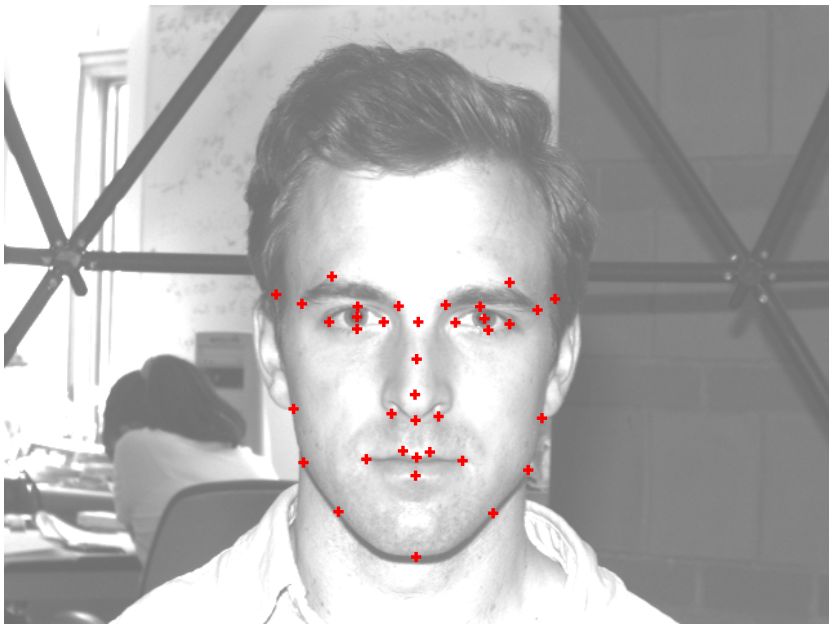


FIGURE 5.1 – Exemple d'un visage labélisé.

5.1.1 Alignement des exemples d'apprentissage

L'analyse procrustéenne est une méthode couramment utilisée pour aligner un ensemble de formes dans un référentiel commun. Elle consiste à minimiser la somme des distances quadratiques de chaque forme à la moyenne ($d = \sum_i \|\mathbf{s}_i - \bar{\mathbf{s}}\|^2$) en fonction des paramètres externes. Cette minimisation est obtenue par la méthode itérative présentée dans la figure 5.2.

1. Translater les N_{ex} formes de manière à ce que leur centre de gravité soit à l'origine.
2. Calculer la moyenne de ces formes $\bar{\mathbf{s}}$ et normaliser $\bar{\mathbf{s}} \leftarrow \bar{\mathbf{s}}_1$.
3. Choisir cette forme comme forme de référence.
4. Aligner chaque exemple d'apprentissage sur cette forme de référence.
5. Calculer la nouvelle estimation de la forme moyenne
6. Aligner la forme moyenne sur la forme de référence et la normaliser.
7. Aller à 4 tant que la moyenne des formes alignées varie significativement par rapport à l'itération précédente

FIGURE 5.2 – Méthode d'alignement procrustéen.

A l'étape 4, toutes les formes sont centrées sur l'origine. L'alignement de deux formes

$\mathbf{s}_1$ et $\mathbf{s}_2$ consiste donc à trouver le coefficient de mise à l'échelle k et l'angle de rotation θ qui minimise $|T_{k,\theta}(\mathbf{s}_1) - \mathbf{s}_2|^2$, la somme des distances entre les points du modèle $\mathbf{s}_2$ et ceux du modèle $\mathbf{s}_1$ ayant subi une rotation et un changement d'échelle. Cootes [2000] propose une méthode simple d'estimation de k et θ. Soit :

$$a = ({\mathbf{s}_1}^T \mathbf{s}_2)/|\mathbf{s}_1|^2 \tag{5.2}$$

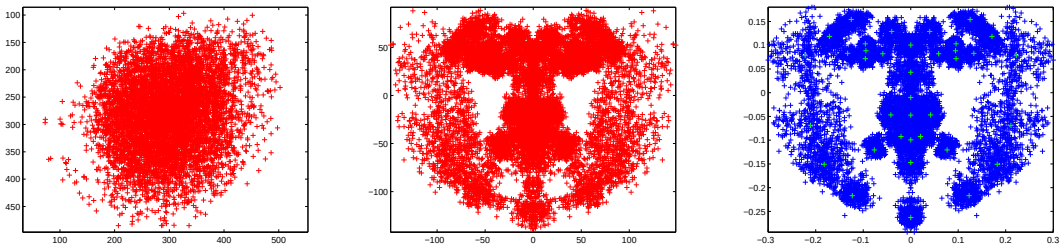
$$b = \left(\sum_{i=1}^{V}(x_{i,1}y_{i,2} - y_{i,1}x_{i,2})\right)/|\mathbf{s}_1|^2 \tag{5.3}$$

L'angle de rotation et le coefficient de mise à l'échelle sont alors donnés par :

$$\theta = \arctan(b/a) \tag{5.4}$$

$$k = \sqrt{a^2 + b^2} \tag{5.5}$$

La figure 5.3 illustre le résultat de la première et de la dernière étape de l'alignement procrustéen.



(a) (b) (c)

FIGURE 5.3 – Exemple d'alignement procrustéen. (a) Ensemble des formes avant alignement. (b) Ensemble des formes centrées à l'origine (étape 1 de la méthode). (c) Ensemble des formes après alignement ; les points verts correspondent à la forme moyenne.

5.1.2 Modélisation des variations de forme

La section précédente décrivait une méthode pour aligner les formes dans un référentiel commun. Cette partie présente un moyen de modéliser la variation des formes dans ce référentiel. L'objectif est d'exploiter les corrélations entre les mouvements des différents points du modèle afin de réduire le nombre de paramètres de déformation du modèle de forme. Par exemple, dans le cas très simple d'une translation suivant un axe, un seul paramètre est nécessaire pour décrire le déplacement de l'ensemble des points du modèle. L'analyse

en Composante Principale, déjà évoquée au chapitre 1.2.1, est une approche classiquement employée pour modéliser des variations de forme. Elle consiste à :

1. Estimer la forme moyenne
$$\bar{\mathbf{s}} = \frac{1}{N_{ex}} \sum_{i=1}^{N_{ex}} \mathbf{s}_i \tag{5.6}$$
2. Estimer la matrice de covariance
$$\mathbf{C} = \frac{1}{N_{ex} - 1} \sum_{i=1}^{N_{ex}} (\mathbf{s}_i - \bar{\mathbf{s}})(\mathbf{s}_i - \bar{\mathbf{s}})^T \tag{5.7}$$
3. Calculer les vecteurs propres $\boldsymbol{\phi}_i$ et les valeurs propres associées λ_i de $\mathbf{C}$. Les valeurs propres sont triées telles que $|\lambda_i| \geq |\lambda_{i+1}|$.

En conservant les N_p premiers vecteurs propres, ont peut approximer n'importe quelle forme de la base d'apprentissage par :

$$\mathbf{s} \approx \bar{\mathbf{s}} + \sum_{i=1}^{N_p} p_i \boldsymbol{\phi}_i \tag{5.8}$$

Cette équation peut s'écrire matriciellement :

$$\mathbf{s} \approx \bar{\mathbf{s}} + \boldsymbol{\Phi}\mathbf{p} \tag{5.9}$$

avec $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1 | \boldsymbol{\phi}_2 | \dots | \boldsymbol{\phi}_{N_p})$ la matrice contenant en colonne les N_p vecteurs propres et $\mathbf{p}$ le vecteur regroupant les paramètres de déformation p_i. On peut ainsi générer différentes instances du modèle en faisant varier les éléments de $\mathbf{p}$. La figure 5.4 illustre une instance du modèle obtenue à partir des exemples de la figure 5.3

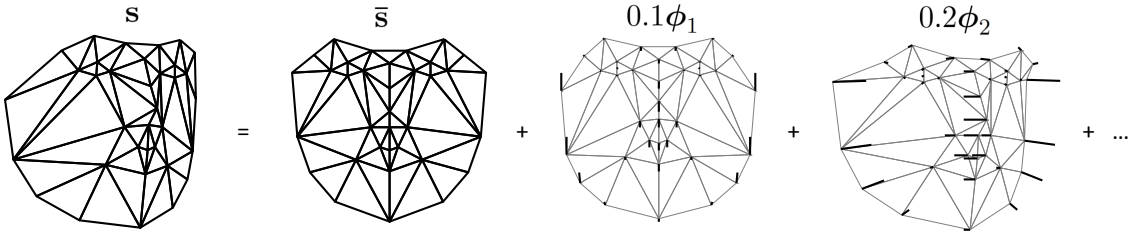


FIGURE 5.4 – Exemple d'instanciation du modèle de forme.

La figure 5.5 représente des exemples générés à partir du modèle de forme. Chaque ligne correspond à la variation d'un paramètre. Les paramètres sont bornés entre $-3\sqrt{\lambda_i}$ (colonne de gauche) et $3\sqrt{\lambda_i}$ (colonne de droite).

Inversement, il est possible de déterminer les paramètres du modèle qui ont permis de générer une forme :

$$\mathbf{p} = \boldsymbol{\Phi}^T(\mathbf{s} - \bar{\mathbf{s}}) \tag{5.10}$$

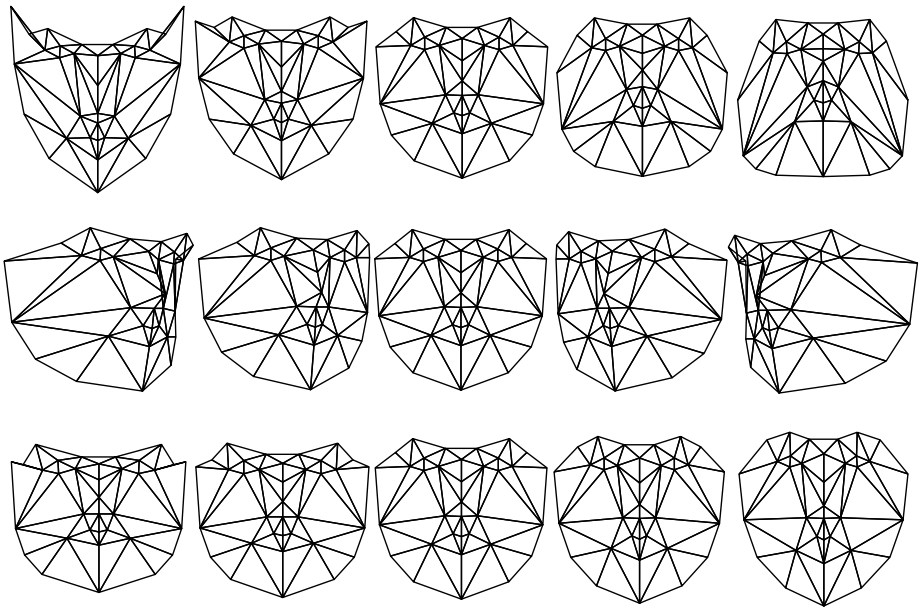


FIGURE 5.5 – Exemples générés à partir des premiers vecteurs propres du modèle de forme.

L'équation 5.9 exprime uniquement les déformations internes du modèle. Pour décrire complètement l'état du modèle dans une image, il faut également définir la transformation géométrique $T(\mathbf{s};\mathbf{q})$. Le vecteur $\mathbf{q} = (t_x, t_y, k, \theta)$ regroupe les paramètres externes que nous avions éliminés lors de l'étape d'alignement. $T(\mathbf{s};\mathbf{q})$ correspond donc à une similitude dans le plan. Par exemple, appliquée au point $\mathbf{x} = (x, y)$ on obtient :

$$T(\mathbf{x};\mathbf{q}) = \begin{pmatrix} k\cos\theta & -k\sin\theta \\ k\sin\theta & k\cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{5.11}$$

Aligner un modèle consistera donc à déterminer les paramètres internes $\mathbf{p}$ et externes $\mathbf{q}$ qui décrivent au mieux la forme d'un visage dans une image inconnue.

5.2 Modèle d'apparence

Le modèle d'apparence vise à caractériser la texture située à l'intérieur du polygone joignant les points de la frontière du modèle de forme. Dans la suite du document, on désignera cette zone comme l'intérieur du modèle de forme. On distingue les deux étapes suivantes :

- Le transfert de texture : pour une position donnée du modèle de forme dans l'image courante, on extrait sa texture et on la transfère vers un modèle de référence ($\bar{\mathbf{s}}$ par exemple). Cette étape a pour effet de rendre la texture indépendante des variations de forme.
- L'extraction de caractéristiques pour décrire la texture transférée.

5.2.1 Transfert de texture

Soit $\mathcal{S}$ l'ensemble des coordonnées des pixels $\mathbf{x} = (x, y)$ contenus à l'intérieur de la forme de référence $\bar{\mathbf{s}}$ et $W(\mathbf{x}; \mathbf{p}, \mathbf{q})$ une fonction de transfert (*warping function* en anglais) du système de coordonnées de la forme de référence vers le système de coordonnées image. Cette fonction dépend des paramètres internes $\mathbf{p}$ et externes $\mathbf{q}$ comme le montre la figure 5.6.

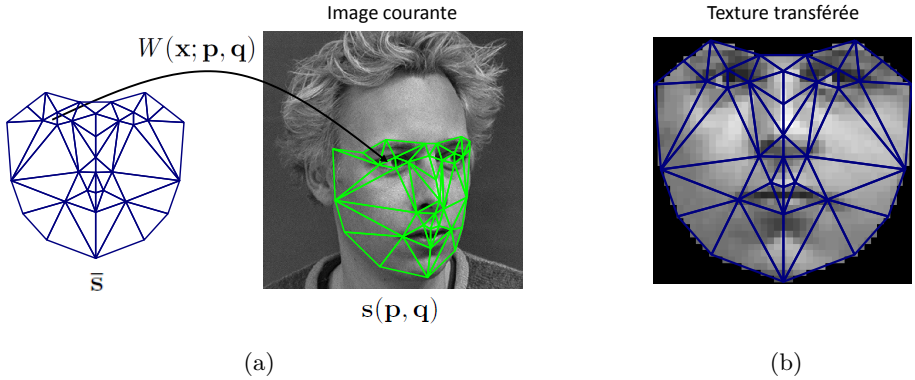


FIGURE 5.6 – Transfert de texture. (a) La fonction de transfert $W(\mathbf{x}; \mathbf{p}, \mathbf{q})$ associe à un pixel $\mathbf{x}$ du modèle de référence $\bar{\mathbf{s}}$, le pixel correpondant dans l'instance $\mathbf{s}(\mathbf{p}, \mathbf{q})$ du modèle dans l'image. (b) $\mathbf{I}(W(\mathbf{x}; \mathbf{p}, \mathbf{q}))$, la texture du modèle $\mathbf{s}(\mathbf{p}, \mathbf{q})$ transférée vers le modèle de référence $\bar{\mathbf{s}}$.

Si l'on définit un maillage triangulaire à partir des points du modèle [1], la fonction de transfert peut être définie comme une transformation affine pour chaque triangle. La figure 5.7 présente une méthode de transfert de texture.

Il est intéressant de noter que les deux premières instructions de l'algorithme dépendent uniquement du modèle de référence. Le triangle englobant et les coordonnées barycentriques de chaque pixel pourront donc être estimés à l'aide d'une table de correspondance précalculée. Par ailleurs, les coordonnées image retournées par la fonction de transfert $W(\mathbf{x}; \mathbf{p}, \mathbf{q})$ ne sont pas à valeurs entières. La valeur du pixel de coordonnée x est donc issue d'une interpolation (le plus souvent bilinéaire).

5.2.2 Descripteurs de texture et prétraitements

Nous utilisons des descripteurs image pour caractériser la texture transférée dans le modèle de référence $\mathbf{I}(W(\mathbf{x}; \mathbf{p}, \mathbf{q}))$. De nombreux descripteurs peuvent convenir. Nous avons choisi les mêmes descripteurs que pour BRM [Wu *et al.*, 2008]. Il s'agit de l'ensemble $\mathcal{F}$ des descripteurs de Haar H, précédemment décrits dans la section 3.4 de ce document.

1. Les triangles peuvent être définis manuellement ou calculés automatiquement en appliquant des algorithmes tels que la triangulation de Delaunay sur la forme de référence $\bar{\mathbf{s}}$

- **Entrée** : une forme $\mathbf{s}(\mathbf{p}, \mathbf{q})$ définie dans l'image $\mathbf{I}$
- **Sortie** : $\mathbf{I}(W(\mathbf{x}; \mathbf{p}, \mathbf{q}))$, la texture du modèle $\mathbf{s}(\mathbf{p}, \mathbf{q})$ transférée vers le modèle de référence $\bar{\mathbf{s}}$
- **Pour** chaque pixel $\mathbf{x}$ de $\mathcal{S}$:
  1. Définir le triangle où $\mathbf{x}$ est situé
  2. Calculer les coordonnées barycentriques de $\mathbf{x}$ dans ce triangle
  3. Calculer les coordonnées de $W(\mathbf{x}; \mathbf{p}, \mathbf{q})$ à partir des coordonnées des sommets du triangle englobant dans l'image courante et des coordonnées barycentriques de $\mathbf{x}$ (*cf.* figure 5.8)
  4. Assigner la valeur $\mathbf{I}(W(\mathbf{x}; \mathbf{p}, \mathbf{q}))$ au pixel de coordonnée $\mathbf{x}$
- **Fin Pour**

FIGURE 5.7 – Algorithme de transfert de texture.

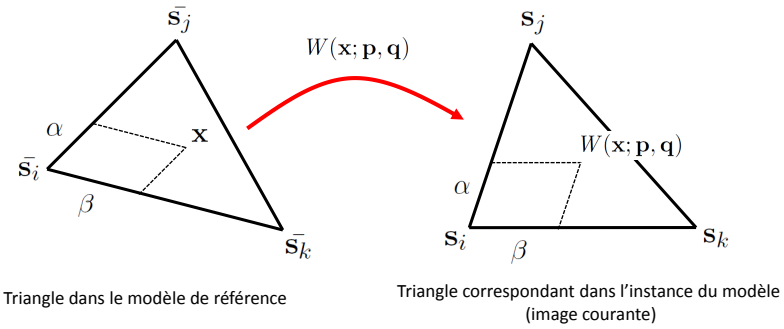


FIGURE 5.8 – Les coordonnées $W(\mathbf{x}; \mathbf{p}, \mathbf{q})$ sont calculées à partir des coordonnées des sommets du triangle englobant dans l'image courante et des coordonnées barycentriques α et β de $\mathbf{x}$.

Le modèle d'apparence correspond donc à une sélection de N_{desc} descripteurs pertinents $\{H_k\}_{i=1}^{N_{desc}}$.

Nous souhaitons par ailleurs normaliser la texture par une égalisation d'histogramme. Toutefois, il n'est pas pertinent d'appliquer l'égalisation sur la texture transférée. En effet, lorsque le modèle n'est pas correctement aligné, la texture du modèle pourra contenir des pixels qui n'appartiennent pas au visage que l'on souhaite segmenter. Ces pixels, intégrés dans le calcul de l'histogramme, peuvent induire de fortes variations d'intensité lumineuse au cours du processus d'alignement. Afin d'éviter ces variations, l'égalisation sera appliquée sur l'image d'origine $\mathbf{I}$, uniquement à partir de l'histogramme des niveaux de gris des pixels du visage détecté (voir figure 5.9). Ainsi la normalisation sera stable au cours du processus d'alignement et indépendante du fond de l'image.

De plus, on économise du temps de calcul en traitant l'image d'origine plutôt que la texture transférée car le prétraitement n'est appliqué qu'une seule fois, à l'initialisation du

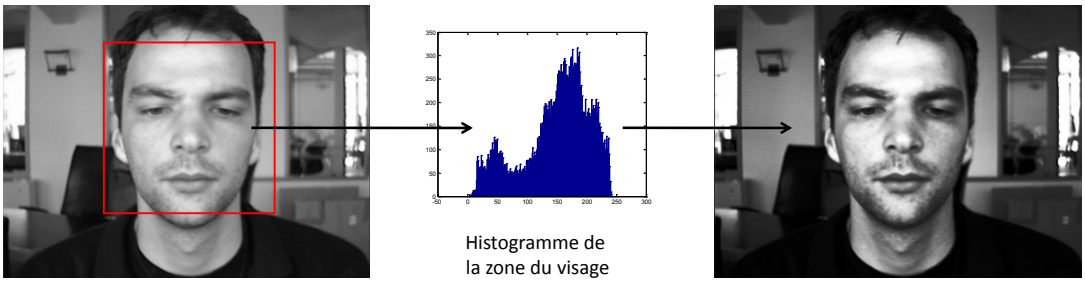


FIGURE 5.9 – Prétraitement : seuls les pixels de la zone du visage sont pris en compte dans la procédure d'égalisation d'histogramme appliquée à toute l'image

processus d'alignement.

5.3 Fonction de coût

Nous venons de définir le modèle de forme et d'apparence de notre méthode d'alignement. Nous présentons dans cette partie, notre fonction de coût adaptée, qui mesurera l'adéquation entre des paramètres du modèle et une image. Nous définirons et nous justifierons dans un premier temps notre choix de fonction de coût idéale. Dans un second temps, nous présenterons le protocole d'apprentissage de cette fonction.

5.3.1 Définition de notre fonction de coût idéale

Soit $\mathbf{b} = (\mathbf{p}^T, \mathbf{q}^T)^T$ le vecteur regroupant les paramètres externes et internes du modèle de forme et $\mathbf{s}_\mathbf{I}^*$ la vérité terrain des points du modèle dans l'image $\mathbf{I}$. Nous définissons notre fonction de coût idéale par :

$$F^*(\mathbf{I}, \mathbf{b}) = \frac{1}{N_{pt}} \|\mathbf{s}(\mathbf{b}) - \mathbf{s}_\mathbf{I}^*\|^2 \tag{5.12}$$

Il s'agit de la distance euclidienne quadratique moyenne entre les points du modèle et leur vérité terrain respective. Cette mesure est également connue sous le terme de *distance quadratique point à point*. Le choix de cette fonction de coût idéale peut sembler surprenant car la fonction n'est pas convexe. En effet, à mesure que l'on s'éloigne de $\mathbf{b}_\mathbf{I}^*$ dans l'espace des paramètres, on n'est pas sûr que la valeur de la fonction de coût continue à croître. La fonction

$$F_{alt}^*(\mathbf{I}, \mathbf{b}) = \|\mathbf{b} - \mathbf{b}_\mathbf{I}^*\| \tag{5.13}$$

est convexe et peut sembler plus judicieuse au premier abord. Nous avons toutefois privilégié la définition (5.12) pour les raisons suivantes :

1. La fonction (5.12) semble empiriquement convexe au voisinage de l'optimum.
2. La distance point à point est un choix naturel puisqu'elle correspond également à une mesure souvent utilisée pour évaluer la qualité d'un alignement.
3. L'impact de la modification d'un paramètre sur le déplacement des points du modèle dépend fortement du paramètre considéré. La première définition de la fonction de coût prend implicitement en compte cette sensibilité en fonction des paramètres.
4. La fonction de coût (5.13) dépend de $\mathbf{b}_\mathbf{I}^*$ qui correspond à une valeur idéale des paramètres. Le vecteur $\mathbf{b}_\mathbf{I}^*$ est défini par :

$$\|\mathbf{s}(\mathbf{b}_\mathbf{I}^*) - \mathbf{s}_\mathbf{I}^*\|^2 = 0 \tag{5.14}$$

Mais rien ne garantit l'unicité ni même l'existence de $\mathbf{b}_\mathbf{I}^*$. Par contre la fonction de coût (5.12) dépend de la vérité terrain $\mathbf{s}_\mathbf{I}^*$ qui est connue.

5.3.2 Apprentissage de la fonction de coût

Apprendre la fonction de coût consiste à apprendre, pour chaque image de la base d'apprentissage, la relation entre l'apparence de la forme pour différentes valeurs de $\mathbf{b}$ et la valeur de la fonction de coût idéale $F^*(\mathbf{I}, \mathbf{b})$ correspondante. Il s'agit donc d'un problème conjoint de régression et de sélection de descripteurs qui peut être traité par BISAR.

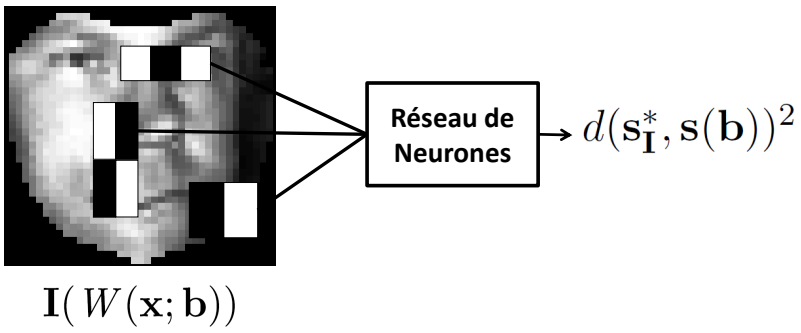


FIGURE 5.10 – L'algorithme BISAR sélectionne les descripteurs qui permettent de prédire au mieux l'erreur d'alignement.

Notre base de données d'apprentissage est constituée de N_{im} images de visages $\mathbf{I}_i$. Chaque image est munie de sa vérité terrain $\mathbf{s}_{\mathbf{I}_i}^*$. On peut calculer le vecteur de paramètres $\tilde{\mathbf{b}}_{\mathbf{I}_i}$ qui ajuste au mieux le modèle à la vérité terrain dans l'image. On définit $\tilde{\mathbf{b}}_\mathbf{I}$ par :

$$\tilde{\mathbf{b}}_\mathbf{I} = \underset{\mathbf{b}}{\operatorname{argmin}} \|\mathbf{s}(\mathbf{b}) - \mathbf{s}_\mathbf{I}^*\| \tag{5.15}$$

Dans le cas de modèles statistiques 2D, on utilise une méthode itérative proposée par Cootes et Taylor [2004, section 4.8]. Elle consiste à estimer successivement les paramètres de translation, rotation et changement d'échelle qui alignent au mieux le modèle à la vérité

terrain, puis à estimer les paramètres internes du modèle. On génère également des exemples plus ou moins bien alignés en perturbant aléatoirement ce vecteur de paramètres $\tilde{\mathbf{b}}_{\mathbf{I}_i}$

$$\mathbf{b}'_{\mathbf{I}_i} = \tilde{\mathbf{b}}_{\mathbf{I}_i} + \mu_j \Delta \mathbf{b}_j \tag{5.16}$$

Le vecteur $\Delta \mathbf{b}_j$ désigne un déplacement dans l'espace des paramètres. Chaque composante du vecteur est issue d'un tirage aléatoire uniforme sur un intervalle dont les bornes, propres à chaque paramètre, sont estimées par une analyse statistique sur la base d'apprentissage. Le scalaire μ_j permet de moduler l'amplitude des déplacements.

Les exemples d'apprentissage sont des couples composés de l'apparence du visage pour différentes instances du modèle d'une part, et de la distance entre ces modèles et la vérité terrain d'autre part (*cf.* figure 5.11).

$$\Big(\mathbf{I}_i(\,W(\mathbf{x}; \tilde{\mathbf{b}}_{\mathbf{I}_i} + \mu_j \Delta \mathbf{b}_j)), d(\mathbf{s}^*_{\mathbf{I}_i}, \mathbf{s}(\tilde{\mathbf{b}}_{\mathbf{I}_i} + \mu_j \Delta \mathbf{b}_j))\Big) \tag{5.17}$$

On choisira les valeurs de μ_j de manière à ce que les valeurs de sortie soient uniformément réparties. Si $\tilde{\mathbf{b}}_{\mathbf{I}} \neq \mathbf{b}^*_{\mathbf{I}}$ on n'aura pas d'exemples d'images transférées pour la bonne position du modèle. Dans le cadre de notre fonction de coût ce n'est pas un problème car on n'a pas besoin de connaître $\mathbf{b}^*_{\mathbf{I}}$. La fonction de transfert étant définie dès que l'on a deux formes (la forme de départ et la forme de destination), on peut générer un exemple d'apprentissage pour une bonne position du modèle à partir de la verité terrain $\mathbf{s}^*_{\mathbf{I}_i}$:

$$\Big(\mathbf{I}_i(\,W(\mathbf{x}; \mathbf{s}^*_{\mathbf{I}_i})), 0\Big) \tag{5.18}$$

Cet exemple est représenté par la première colonne de la figure 5.11

L'algorithme BISAR (*cf.* section 3.3) sélectionnera les descripteurs de l'apparence (des filtres de Haar) qui permettent au réseau de neurones (un GRNN dans notre cas) de prédire au mieux la distance entre le modèle et la vérité terrain. Ce principe est illustré par la figure 5.10.

5.4 Optimisation des paramètres

Nous pensons que notre fonction de coût idéale (equation (5.12)) est convexe au voisinage de la bonne position. Nous espérons alors que notre fonction apprise héritera de cette propriété et sera donc facile à minimiser. Notre choix s'est porté sur la méthode de recherche exhaustive et dirigée (E&D) proposée par Dornaika et Ahlberg [2006]. Il s'agit d'une méthode numérique du première ordre dans laquelle on distingue deux phases :

1. *La phase d'exploration.* Partant d'une solution initiale $\mathbf{b}^{(0)}$, on modifie à chaque itération $l = 1, \ldots, N_{iter}$, le paramètre qui améliore le plus l'alignement. On teste chaque paramètre i indépendamment en évaluant la fonction de coût pour N_{ech} valeurs de ce paramètre échantillonnées de façon régulière sur l'intervalle $\Big[\mathbf{b}^{(l)}(i) - \frac{\delta_i}{2},\ \mathbf{b}^{(l)}(i) + \frac{\delta_i}{2}\Big]$. Le vecteur $\mathbf{b}^{l+1}$ ne diffère du vecteur $\mathbf{b}^l$ que pour une composante, correspondant à la valeur du paramètre pour laquelle la valeur de la fonction de coût était minimum.

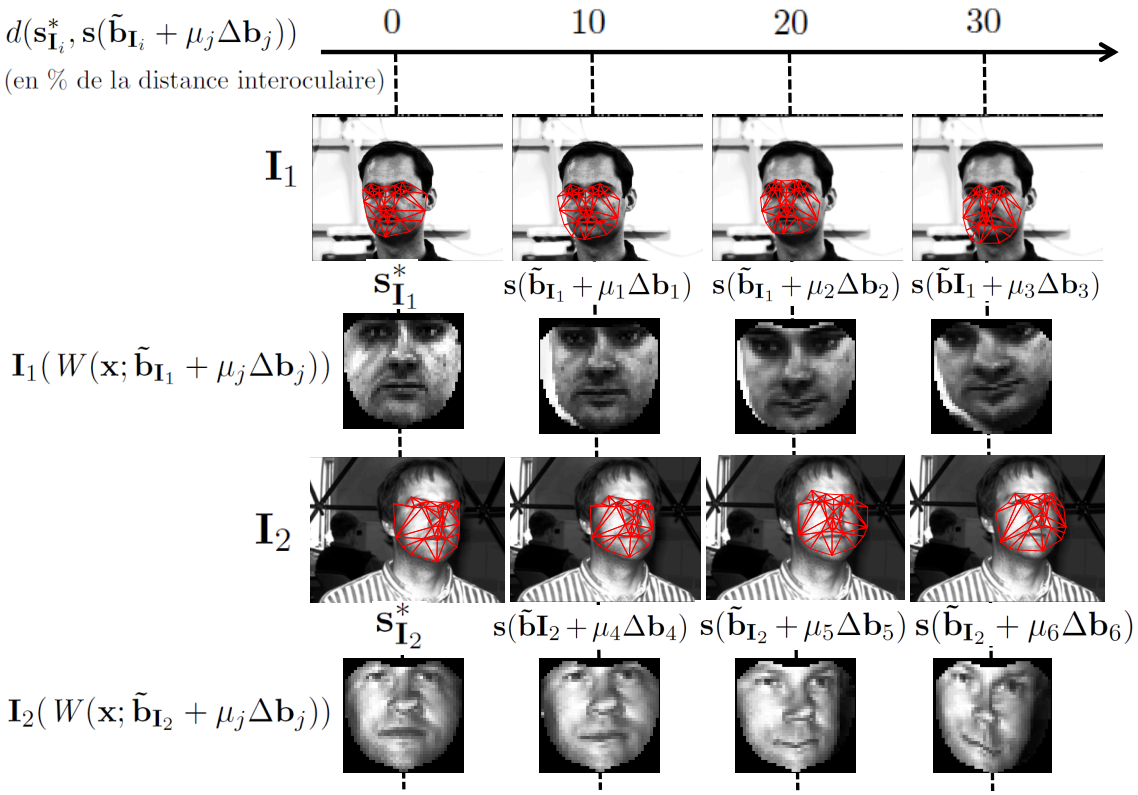


FIGURE 5.11 – Exemples d'apprentissage. Pour chaque image de la base d'apprentissage (ici, deux images ont été représentées), on perturbe les paramètres de forme de manière à ce que le modèle soit de plus en plus éloigné de sa bonne position. Un exemple d'apprentissage est constitué de l'apparence d'une instance du modèle dans l'image et la distance de ce modèle à la vérité terrain

Le processus s'arrête sur la solution $\mathbf{b}^{(L)}$ lorsque la valeur de la fonction de coût ne diminue plus ou lorsque le nombre d'itération maximal N_{iter} a été atteint.

2. *La phase d'affinement.* On considère que la solution $\mathbf{b}^{(L)}$ désigne la direction dans laquelle la pente de la fonction de coût est la plus forte. La seconde étape consiste à trouver le meilleur jeu de paramètres $\hat{\mathbf{b}}$ tel que

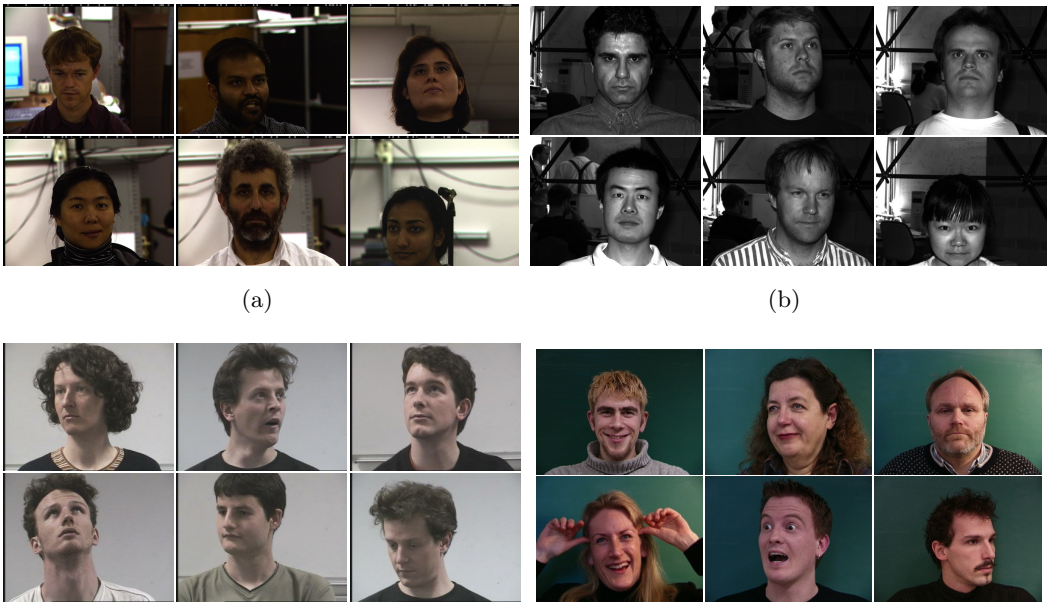
$$\hat{\mathbf{b}} = \underset{\mu}{\operatorname{argmin}} \left(F(\mathbf{I}, \mathbf{s}(\mathbf{b}(\mu))) \right) \text{ avec } \mathbf{b}(\mu) = \mathbf{b}^{(L)} + \mu(\mathbf{b}^{(L)} - \mathbf{b}^{(0)}) \tag{5.19}$$

Nous venons de décrire les différentes parties de notre méthode d'alignement. Nous allons à présent évaluer ses performances au travers de différentes expérimentations.

5.5 Evaluation

5.5.1 Bases de données

Pour évaluer notre méthode, nous avons collecté 698 images issues des bases de données publiques PIE [Sim *et al.*, 2003], Yale [Georghiades *et al.*, 2001], IMM [Nordstrøm *et al.*, 2004] et Pointing [Gourier *et al.*, 2004a]. La figure 5.12 montre quelques échantillons de ces bases.



(a) (b)

(c) (d)

FIGURE 5.12 – Echantillons des bases de données. (a) Base PIE. (b) Base Yale. (c) Base Pointing 04. (d) Base IMM.

Les images ont été réparties en 5 groupes. Le premier groupe contient 150 images de 50 individus de la base PIE et 40 images de 9 individus de la base Yale. Le deuxième groupe rassemble 60 images de la base PIE et 40 images de la base Yale. Les individus sont les mêmes que dans le groupe 1 mais les prises de vues sont différentes. Le groupe 3 contient 69 images de 17 individus de la base PIE et 9 images d'un même individu de la base Yale. Les individus de ce groupe ne sont pas dans les deux premiers groupes. Le groupe 4 contient 90 clichés des 15 individus de Pointing et le groupe 5 rassemble la totalité des images des 40 individus de la base IMM. Le groupe 1 et 2 sont respectivement utilisés comme base d'apprentissage et de validation. Les trois autres groupes sont les bases de test. Elles permettent de tester deux niveaux de généralisation : le groupe 3 teste la sensibilité de la méthode aux changements d'identité et les deux autres groupes évaluent sa capacité à généraliser sur de nouvelles bases de données (changement d'identité et de conditions de prise de vues).

TABLE 5.1 – Répartition des images dans les différents ensembles de données. La valeur entre parenthèses indique le nombre d'individus différents

	PIE	Yale	Pointing	IMM
Variations	Pose, arrière-plan	Pose, arrière-plan	Pose	Pose, expression
Groupe 1	150 (50)	40 (9)	-	-
Groupe 2	60 (50)	40 (9)	-	-
Groupe 3	69 (17)	9 (1)	-	-
Groupe 4	-	-	90 (15)	-
Groupe 5	-	-	-	240 (40)

5.5.2 Mesure de performance

On peut mesurer la qualité d'un alignement de différentes façons. Matthews et Baker [2004] calculent par exemple un taux de convergence. Ils considèrent que le modèle a convergé lorsque la distance euclidienne moyenne entre les points du modèle et la vérité terrain est inférieure à un pixel. Cette mesure ne donne qu'une vision très réduite de la qualité de l'alignement (quid de la distribution des erreurs pour les modèles qui n'ont pas convergé ?). De plus, le seuil de convergence de 1 pixel est fixé arbitrairement. On pourrait également mesurer une erreur dans l'espace des paramètres mais ce type de mesure est difficilement exploitable car chaque paramètre a un impact variable sur le déplacement des points.

La mesure de performance que nous avons retenue dans notre évaluation est la distance euclidienne moyenne entre les points du modèle dans l'image et leur vérité terrain, normalisée par la distance interoculaire d_{io}.

$$\varepsilon_{ali}(\mathbf{s}) = \frac{1}{d_{io} * N_{pt}} \sum_i \|\mathbf{s}_i^* - \mathbf{s}_i\|\| \tag{5.20}$$

Dans la suite de ce document, nous ferons référence à cette mesure sous le terme d'erreur d'alignement. Il s'agit certainement de la mesure la plus naturelle et la plus couramment utilisée pour évaluer la précision d'un alignement. De plus, cette mesure est cohérente avec l'objectif d'un alignement tel que nous l'avons formulé (équation (4.1) du chapitre 4). On peut toutefois noter quelques inconvénients ; cette erreur dépend de la vérité terrain qui est définie manuellement. Elle intègre donc les erreurs d'annotation de la base qui peuvent être importantes pour certains points caractéristiques [Mercier *et al.*, 2006]. Il se peut également qu'il n'y ait pas de solution admissible $\mathbf{b}$ telle que $d(\mathbf{s}(\mathbf{b}), \mathbf{s}^*) = 0$. L'erreur d'alignement ne dissocie pas les erreurs liées à la méthode d'alignement de celles liées aux limitations du modèle de forme.

5.5.3 Apprentissage

Le modèle de forme est un maillage de 37 points définissant 53 triangles. Il a été appris sur le groupe 1. Pour augmenter artificiellement le nombre d'exemples, on applique une réflexion horizontale aux images de la base et on ajoute à l'ensemble d'apprentissage les points issus de ces nouveaux visages. Cela a pour effet d'augmenter le pouvoir d'expression du modèle et de mieux décorréler les variations liées aux rotations horizontales des variations liées aux rotations verticales. On conserve les 12 premiers vecteurs propres correspondant à 95% de l'inertie cumulée.

Le modèle d'apparence est appris sur les 190 images du groupe 1. Les groupes 2 sert de base de validation. Pour chaque image, on génère aléatoirement 10 exemples de synthèse avec des erreurs d'alignement qui varient de 5 à 30% de la distance interoculaire. La texture du visage est transférée vers un modèle de 40 pixels de large. On définit 14909 descripteurs de Haar potentiels pour caractériser cette texture transférée. L'algorithme d'apprentissage BISAR effectue 300 itérations et utilise la stratégie de repondération sans mémoire (*cf.* équation (3.19) page 74). Nous avons conservé les 225 premiers descripteurs sélectionnés ; après cette valeur l'erreur en apprentissage continue à diminuer alors que l'erreur sur la base de validation stagne (cf. figure 5.13).

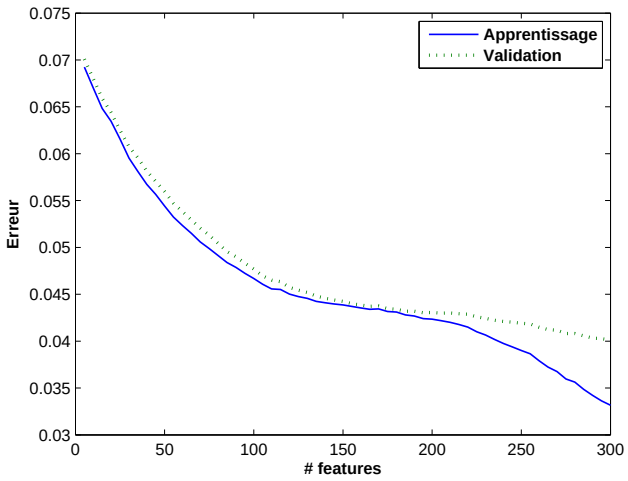
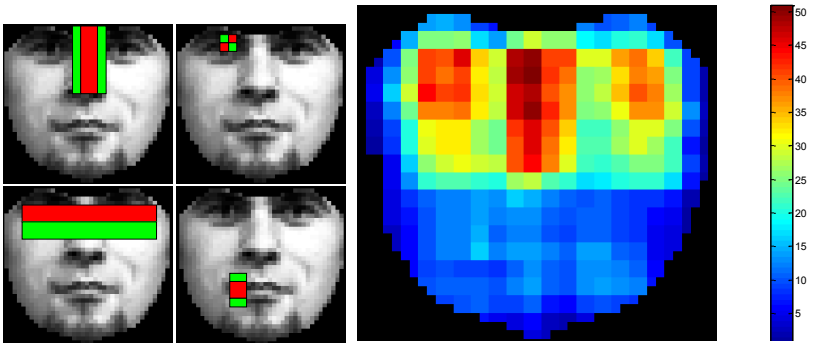


FIGURE 5.13 – Evolution de l'erreur sur la base d'apprentissage et de test au cours des itérations.

La figure 5.14 montre le support des premières ondelettes sélectionnées ainsi que la distribution spatiale des 225 descripteurs. On peut remarquer que les descripteurs se concentrent sur la bouche, les yeux et le nez et plus particulièrement sur ces deux derniers éléments. Ce choix semble pertinent puisqu'il s'agit des zones du visage qui présentent de fortes variations de contraste auxquelles les ondelettes de Haar sont sensibles. On retrouve des répartitions similaires dans les méthodes BAM et BRM.



(a) (b)

FIGURE 5.14 – Descripteurs. (a) Les 4 premiers descripteurs selectionnés. (b) Densité spatiale des 225 descripteurs.

5.5.4 Méthode de comparaison

5.5.4.1 Choix de la méthode

Pour valider notre approche, nous comparons nos résultats à ceux obtenus avec un *Boosted Ranking Model* (BRM Wu *et al.* [2008]). Le tableau 5.2 résume et compare les principaux éléments de BRM et de BiBAM.

TABLE 5.2 – Comparaison entre BRM et BiBAM

	BRM	**BiBAM**
Modèle de forme	Modèle statistique 2D	
Modèle d'apparence	Filtres de Haar sur la texture transférée	
Fonction de coût	Somme de classifieurs faibles appris par RankBoost	Réseau de neurones dont les entrées sont les valeurs des descripteurs sélectionnés par BISAR
Optimisation	Descente de gradient analytique proche de la méthode de composition inverse de Matthews et Baker [2004]	Recherche exhaustive et dirigée (E&D) de Dornaika et Ahlberg [2006]

Nous avons choisi cette méthode d'alignement car :

- Elle est *récente*.
- Elle est *performante* : Wu *et al.* [2008] ont montré que BRM donne de meilleurs résultats que les méthodes orientées AAM [Cootes *et al.*, 2001; Matthews et Baker, 2004].
- Elle est conceptuellement *proche de notre méthode* : on pourra comparer plus finement les différentes parties de notre algorithme, en particulier la fonction de coût. La comparaison sera également plus équitable car le choix des paramètres tels que

l'ensemble des descripteurs potentiels ou la résolution de la texture transférée sera le même pour les deux méthodes.

5.5.4.2 Apprentisage de BRM

Dans nos expériences, nous avons cherché à ce que le protocole expérimental et les paramètres des deux méthodes soient le plus proche possible. Ainsi, nous avons utilisé le même modèle de forme et le même ensemble de descripteurs. Le modèle d'apparence est appris sur les mêmes exemples d'apprentissage (groupe 1). Pour chaque exemple, 60 exemples de synthèse ont été générés, correspondant à 10 vecteurs de perturbation $\Delta\mathbf{b}_u$ et à 6 intensités v par déplacement (*cf.* équation 4.15 page 105). Dans leur article, Wu *et al.* [2008] sélectionnent 50 descripteurs. Nous avons décidé de mener l'expérience 3 (section 5.5.7) avec 50, 100, 200 et 300 descripteurs. Les résultats que nous présentons dans la suite de ce chapitre ont été obtenus avec 200 descripteurs et correspondent aux meilleurs résultats obtenus sur les images du groupe 3, 4, et 5. Nous avons ajusté un paramètre directement sur les bases de test pour favoriser légèrement la méthode de comparaison.

5.5.5 Expérience 1 : nombre de descripteurs

Cette première expérience évalue l'impact du nombre de descripteurs sur la qualité de l'alignement. Elle consiste à initialiser le modèle dans une image, à exécuter l'algorithme d'alignement et à évaluer le résultat. Pour initialiser le modèle, on part de la vérité terrain et on perturbe aléatoirement ses paramètres internes et externes. L'amplitude de la perturbation est choisie de manière à ce que l'erreur d'alignement à l'initialisation soit successivement de 5, 10, 15, 20, 25 et 30%. On répète ce processus pour chaque image de la base de validation (groupe 2). La figure 5.15 montre l'erreur d'alignement en fonction de l'erreur d'initialisation. Chaque courbe correspond à un nombre de descripteurs différent

Ce test montre que l'augmentation du nombre de descripteurs améliore la qualité de l'alignement. Toutefois l'amélioration observée en passant de 225 descripteurs à 300 est faible. Nous avons donc décidé de ne conserver que 225 classifieurs pour limiter les risques d'une mauvaise généralisation sur les bases de test. De plus, cette expérience conforte le choix du nombre de descripteurs que nous avions fait lors de l'apprentissage.

5.5.6 Expérience 2 : taux de bons classements

Dans cette deuxième expérience, on s'intéresse uniquement à la fonction de coût. On initialise aléatoirement le modèle dans l'image $\mathbf{I}$ avec le jeu de paramètres $\mathbf{b}_0$ et on évalue le score d'alignement avec notre fonction $F(\mathbf{I}, \mathbf{s}(\mathbf{b}_0))$. On déplace ensuite le modèle le long de la direction $\tilde{\mathbf{b}}_\mathbf{I} - \mathbf{b}_0$ de manière à obtenir une variation de l'erreur d'alignement $\Delta\varepsilon$. Autrement dit, on trouve μ tel que $d\left(\mathbf{s}(\mathbf{b}_0), \mathbf{s}(\mu(\mathbf{b}_0 - \tilde{\mathbf{b}}_\mathbf{I}))\right) = \Delta\varepsilon$. On évalue l'alignement pour cette nouvelle position. On considère que le classement est bon lorsque le score retourné

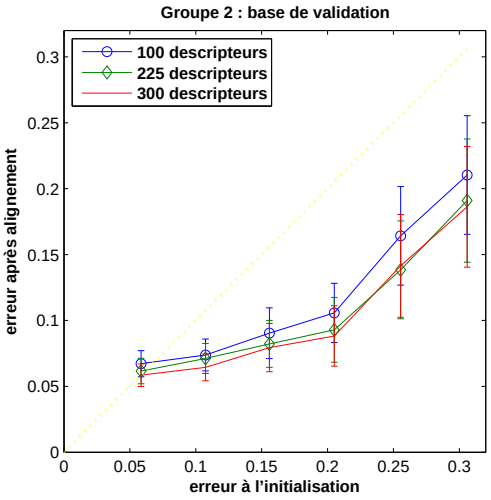


FIGURE 5.15 – Résultats de l'alignement par BiBAM pour différents nombres de descripteurs

par la fonction de coût est inférieur au score précédent. On calcule ainsi un taux de bons classements correspondant au pourcentage de bons classements par rapport au nombre total de paires testées. On réalise 50 évaluations par image pour des modèles initialisés de plus en plus loin de la vérité terrain (jusqu'à 30% de la distance interoculaire). Le tableau compare les résultats obtenus par BRM et BiBAM sur les différentes bases de données pour différentes variations de l'erreur d'alignement $\Delta\varepsilon$

TABLE 5.3 – Comparaison du taux de classement

	BRM			**BiBAM**		
$\Delta\varepsilon$	2.5%	5%	10%	2.5%	5%	10%
Groupe 1	86.3	90.6	90.7	93.6	97.0	97.2
Groupe 2	88.0	90.7	91.0	93.9	97.0	97.2
Groupe 3	87.8	90.6	90.8	93.1	97.0	97.1
Groupe 4	84.7	90.1	89.5	90.0	96.5	95.5
Groupe 5	89.8	91.1	92.4	96.7	97.2	98.0

On remarque tout d'abord que notre méthode améliore substantiellement les résultats obtenus par BRM. On remarque également que les deux méthodes affichent de bons résultats et qu'elles sont capables de généraliser sur des images très différentes de celles de la base d'apprentissage. A noter toutefois les performances légèrement moins bonnes sur la quatrième base de données. Notre dernière remarque porte sur la sensibilité de la fonction de coût qui obtient d'aussi bonnes performances pour des perturbations de 10% que de 5%. Par contre, elle détecte plus difficilement les très faibles variations (de l'ordre de 2.5%). Cette limitation est probablement due à la faible résolution de la texture transférée puisqu'un déplacement de 2.5% de la distance interoculaire représente une variation d'un demi

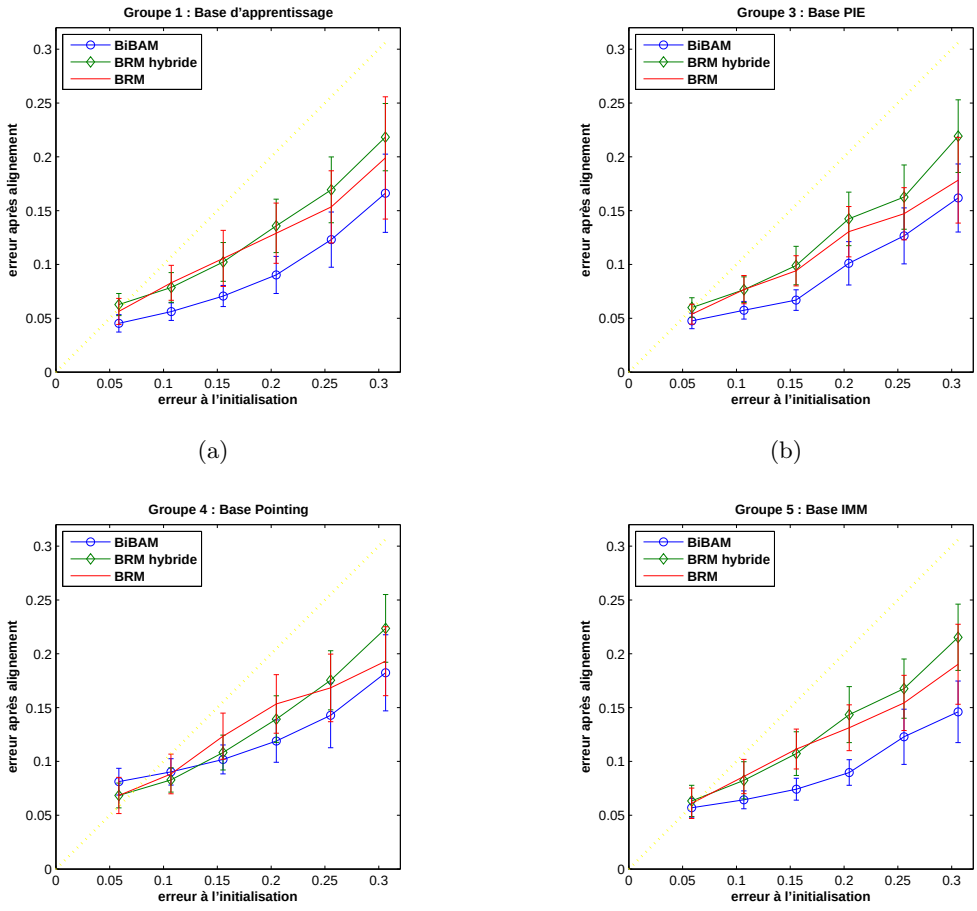
pixel environ pour un visage de 40 pixels de large.

Par ailleurs, cette expérience permet de vérifier que l'erreur estimée par la fonction de coût diminue lorsque le modèle se rapproche de la vérité terrain, mais rien ne garantit qu'il n'existe pas une autre direction dans l'espace des paramètres pour laquelle l'erreur diminuerait encore plus rapidement. Pour évaluer empiriquement la convexité de la fonction, il faudrait explorer le voisinage de chaque position du modèle dans l'espace des paramètres. En ne considérant que 3 positions par paramètres, le nombre d'évaluations nécessaires serait de $3^{N_{param}} * N_{im} * N_{pos}$ avec N_{im} le nombre d'images de la base d'évaluation, N_{pos} le nombre de positions du modèle par image et N_{param} le nombre de paramètres internes et externes. Cette expérience est difficilement réalisable et nous avons donc choisi d'introduire un biais en nous déplaçant toujours vers la bonne position.

5.5.7 Expérience 3 : rayon de convergence

L'expérience 3 utilise le même protocole expérimental que l'expérience 1. Elle consiste à déplacer le modèle de sa bonne position en perturbant aléatoirement ses paramètres. On compare alors la distance du modèle à la vérité terrain à la fin du processus d'alignement par rapport à sa distance à l'initialisation. Dans la méthode BRM, seuls les paramètres internes sont pris en compte dans le processus d'optimisation. Dans cette expérience nous n'avons donc perturbé que les paramètres internes du modèle pour pouvoir nous comparer à BRM. Nous avons également évalué l'alignement en combinant la fonction de coût de BRM avec la méthode de recherche E&D utilisée dans BiBAM. On pourra ainsi voir l'influence de la méthode d'optimisation sur les résultats. On pourra comparer la performance des fonctions de coût de BRM et de BiBAM pour une même méthode d'optimisation. Cette évaluation permettra également de tester la performance de cette méthode hybride car nous l'utiliserons dans les évaluations qui nécessitent d'utiliser les paramètres externes (dans l'expérience 4 par exemple).

On remarque que notre méthode donne de meilleurs résultats que BRM sur la plupart des évaluations. Pour des initialisations loin de la vérité terrain (supérieures à 15%), BiBAM réduit l'erreur de 50% en moyenne par rapport à l'erreur à l'initialisation, alors que l'amélioration apportée par BRM se situe autour de 30%. BiBAM généralise très bien sur les bases de test 3 et 5. Mais sur la base Pointing, on retrouve les difficultés rencontrées lors de l'expérience précédente, en particulier pour de faibles déplacements. On remarque, par ailleurs, que la méthode d'optimisation a une légère influence sur les résultats de l'alignement puisque la méthode BRM originale donne des résultats sensiblement meilleurs que la méthode BRM hybride. Ce résultat est encourageant puisque, d'une part, il confirme que notre fonction de coût est plus performante que celle de BRM (indépendamment de la méthode d'optimisation) et d'autre part, il montre que le choix d'une méthode d'optimisation adaptée peut améliorer les résultats.



(a) (b)

(c) (d)

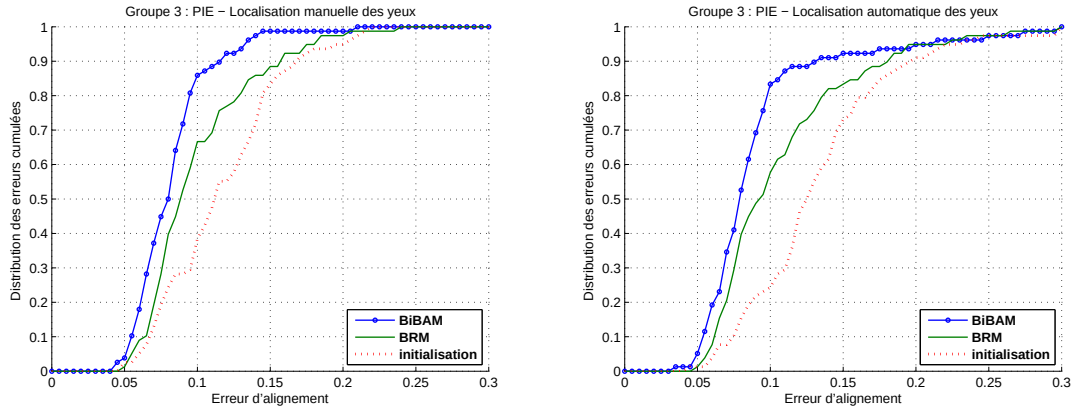
FIGURE 5.16 – Erreur d'alignement en fonction de l'erreur à l'initialisation

5.5.8 Expérience 4 : Dans un cas réel

Dans cette dernière expérience, on initialise le modèle à partir de la position des yeux. Cela consiste à ajuster les paramètres externes de manière à ce que les points du modèle qui correspondent aux centres des yeux soient localisés sur le centre des yeux dans l'image. Par ailleurs, les paramètres internes sont initialisés à 0.

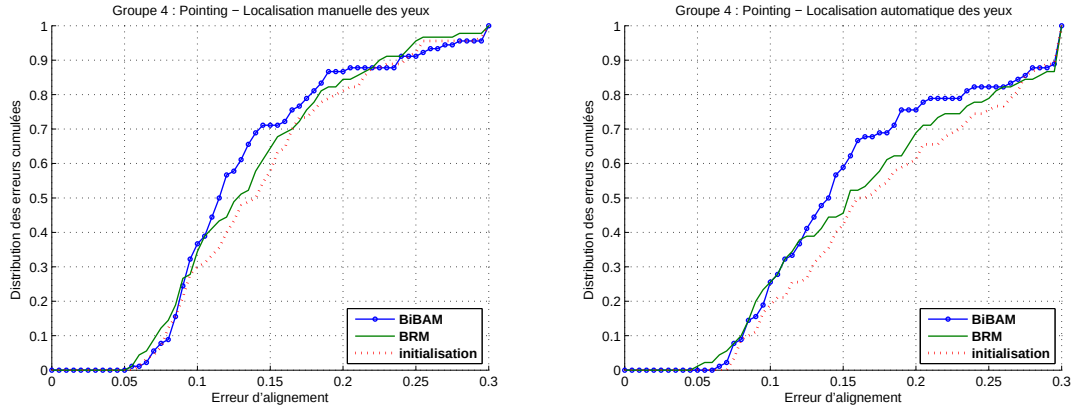
Nous avons testé l'alignement en utilisant la position des yeux donnée par la vérité terrain et celle donnée par une méthode automatique de détection de la pupille [Milgram *et al.*, 2005]. Cette dernière expérience correspond à une situation réelle d'alignement car la vérité terrain n'est plus utilisée pour initialiser le modèle.

Les graphiques de la figure 5.17 illustrent les résultats obtenus sur les différentes bases de test avec initialisation manuelle (colonne de gauche) et automatique (colonne de droite). L'axe horizontal représente l'erreur d'alignement du modèle et l'axe vertical représente le



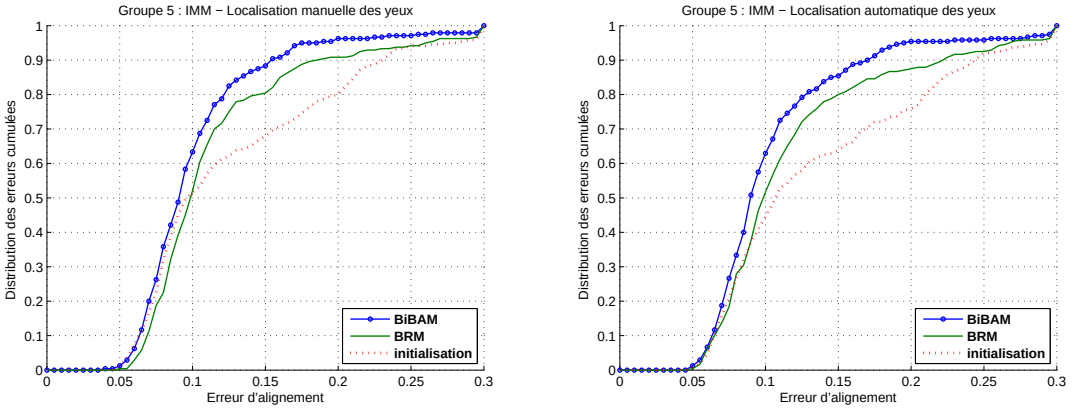
(a)

(b)



(c)

(d)



(e)

(f)

FIGURE 5.17 – Distributions des erreurs d'alignement pour un modèle initialisé avec la vérité terrain des yeux (a,c,e) ou par une méthode automatique (b,d,f)

pourcentage cumulé des images ayant une erreur donnée. La courbe bleue et la courbe verte correspondent respectivement aux résultats de BiBAM et BRM. La courbe en pointillé rouge illustre la distribution des erreurs à l'initialisation. Les figures 5.17(a), 5.17(b), 5.17(e) et 5.17(f) montrent que BRM et BiBAM améliorent nettement la position initiale du modèle. La méthode BiBAM donne de meilleurs résultats que BRM sur les deux bases PIE et IMM. Par contre BRM et BiBAM n'améliorent que très légèrement les estimations initiales sur la base Pointing.

5.6 Discussion et perspectives

Nous avons proposé dans ce chapitre une nouvelle méthode pour aligner un modèle dans une image. A partir d'un ensemble d'images de visages munis de leur vérité terrain, on génère des instances de modèles plus ou moins bien positionnées dans l'image. On apprend ensuite, à l'aide de notre méthode BISAR, la relation entre l'apparence du modèle et son éloignement par rapport à la vérité terrain. Cette mesure constitue la fonction de coût que l'on cherchera à minimiser. Les expérimentations que nous avons menées ont montré l'importance de cette fonction dans le processus d'alignement. Elles ont également mis en évidence les bonnes performances de notre approche puisque celle-ci améliore sensiblement les résultats obtenus par BRM. Notre méthode a, de plus, montré une bonne capacité à généraliser sur des visages qui n'ont pas été vus en apprentissage. On observe une bonne robustesse aux variations d'identité, de morphologie, d'expression et de pose.

Les différentes évaluations ont toutefois révélé certaines limitations :

- Le rayon de convergence : lorsque le modèle est initialisé trop loin, la méthode d'alignement n'est pas capable de l'amener vers sa bonne position. Cette limitation est principalement liée à l'intensité maximale des déformations que nous avons générées dans la base d'apprentissage. Si l'initialisation dépasse cette intensité, BiBAM n'aura pas d'exemple auquel se rattacher pour prédire l'erreur d'alignement.
- La précision : elle est fonction de la sensibilité de la fonction de coût. Si la fonction est sensible, elle sera capable d'identifier des améliorations pour de faibles déplacements du modèle. Cela nécessite une bonne résolution de la texture transférée. Si la résolution est trop faible, de petits déplacements du modèle n'auront pas d'impact sur la variation des niveaux de gris de la texture transférée. Pour que la fonction de coût soit sensible il faut également que la base de données contienne des exemples d'apprentissage avec des erreurs proches d'alignement. Cette contrainte s'oppose donc à celle que nous venons de formuler pour la portée de la méthode. Le nombre d'exemples de la base d'apprentissage étant limité, il faut impérativement faire un compromis entre la précision et le rayon de convergence du modèle.
- La généralisation : BiBAM, tout comme BRM, ont montré des difficultés à aligner des modèles sur les images de la base Pointing. Les conditions de prise de vue (éclairage, compression de l'image...) doivent certainement être trop éloignées de celles des exemples d'apprentissage et la fonction de coût n'est plus capable de donner une

évaluation fiable de l'erreur l'alignement.

Certaines pistes méritent d'être explorées pour répondre à ces problématiques. On pourra par exemple envisager une stratégie de type multi-résolution qui minimise différentes fonctions de coût successivement. Pour les premières itérations on utiliserait une fonction de coût avec une faible résolution et un large rayon de convergence, alors que les dernières itérations feraient appel à une fonction de coût très sensible pour ajuster finement le modèle.

On pourrait également introduire une stratégie de *bootstrapping* : on génère de nombreux exemples à partir de modèles plus ou moins bien alignés et l'on réapprend la fonction de coût en incorporant à l'ensemble d'apprentissage, les exemples qui ont été les plus mal estimés.

On pourrait également introduire un biais dans la création des exemples d'apprentissage, en favorisant les exemples qui ont une forte probabilité d'être rencontrés pendant la phase d'alignement. Pour générer de tels exemples, on peut prendre des échantillons le long de la direction définie par la position du modèle bien aligné dans l'image et la position du modèle à l'initialisation (définie avec la position des yeux par exemple). Cette méthode donnerait certainement des exemples plus pertinents que ceux donnés par une direction totalement aléatoire dans l'espace des paramètres. Cette stratégie est toutefois dépendante de l'étape d'initialisation du modèle.

Chapitre 6

De l'alignement à la pose

La phase d'alignement du chapitre précédent a permis de localiser des points caractéristiques du visage dans l'image. Dans cette partie, nous abordons le passage de cette information 2D à l'information 3D de la pose de la tête. Nous débuterons ce chapitre par un aperçu des méthodes rencontrées dans la littérature pour traiter ce problème. Nous présenterons ensuite la solution que nous avons utilisée, ainsi que les résultats obtenus sur des données de synthèse et des images réelles.

6.1 Aperçu des méthodes

Nous distinguons les méthodes orientées apprentissage qui modélisent la relation entre l'information apportée par le modèle 2D et la pose, des méthodes géométriques qui exploitent les contraintes géométriques entre des points caractéristiques du visage.

6.1.1 Méthodes orientées apprentissage

Elles visent à apprendre une relation directe entre l'information du modèle aligné et la pose correspondante. Il peut s'agir par exemple de la position des points dans l'image. On peut alors faire appel à un outil de régression pour apprendre la relation entre les points du modèle $\mathbf{s} \in \mathbb{R}^{N_{pt}}$ et le vecteur $\boldsymbol{\theta} = (\alpha, \beta, \phi) \in \mathbb{R}^3$, regroupant les valeurs du triplet d'angles *pan*, *tilt* et *roll*. Ma *et al.* [2006b] montrent que l'utilisation d'une RVM (*Relevance Vector Machine*) obtient de meilleurs résultats que d'autres méthodes de régression telles que les Analyses Canoniques des Correspondances à Noyaux (*Kernel CCA*) ou les *Support Vector Regression* [SVR, Drucker *et al.*, 1996].

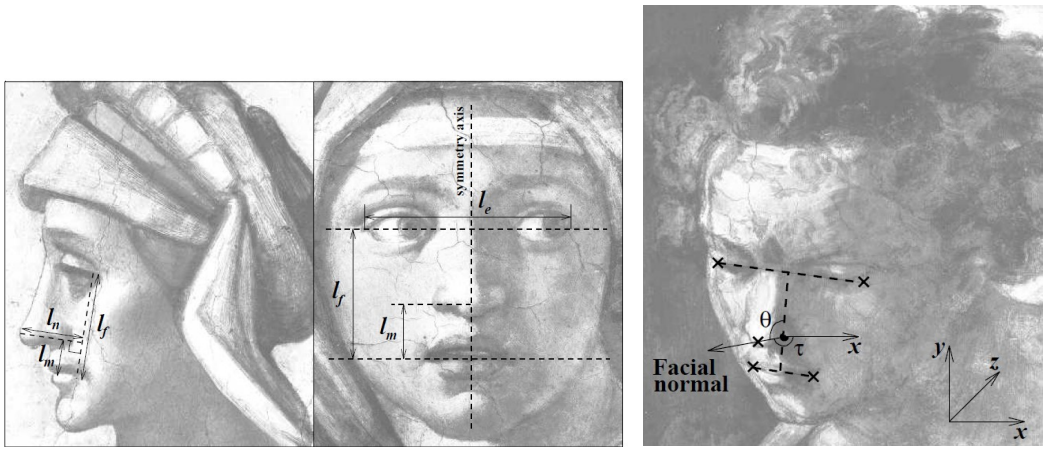
D'autres travaux se sont intéressés à l'information contenue dans les paramètres de déformations internes du modèle pour inférer la pose. Nous avons pu remarquer dans la figure 5.5 de la page 115 que les premiers modes de déformation du modèle encodaient les rotations hors-plan du visage, *pan* et tilt. Cootes *et al.* [2002] estiment, par une régression linéaire, la valeur de l'angle *pan* à partir des paramètres de forme et d'apparence d'un modèle AAM.

Wu et Trivedi [2008] utilisent les *Elastic Bunch Graph* [EBG, Wiskott *et al.*, 1997] pour localiser des points caractéristiques du visage (commissures des yeux, de la bouche...). Pour l'estimation de pose, un EBG est créé pour chaque pose discrète. Chaque EBG est testé

pour une nouvelle image et le modèle qui s'adapte le mieux dans l'image (*i.e.* qui obtient le meilleur score de la fonction de coût) définit la pose du visage. L'inconvénient de cette méthode est que la pose estimée est discrète et une augmentation de la précision implique une augmentation du nombre d'EBG à aligner.

6.1.2 Méthodes géométriques

Elles déterminent la pose de la tête à partir des contraintes géométriques explicites entre différents points caractéristiques du visage. Les méthodes les plus simples n'utilisent que quelques points du visage. Gourier *et al.* [2004b] s'intéressent par exemple à la position des yeux par rapport au centre de la tête et détermine l'angle de rotation horizontale à l'aide de règles trigonométriques simples. Pour évaluer l'angle de rotation verticale, d'autres attributs du visage sont pris en compte tels que la bouche [Nikolaidis et Pitas, 2000], le nez [Kaminski *et al.*, 2006] ou les deux [Gee et Cipolla, 1994]. Les procédés pour extraire l'orientation du visage à partir de ces positions sont très variés. Gee et Cipolla [1994] utilisent par exemple la position des yeux et les commissures de la bouche pour définir le plan de symétrie du visage. Considérant que le ratio $R_m = l_f/l_m$ (*cf.* figure 6.1) est connu et constant pour tous les visages, ils retrouvent la position de la base du nez dans l'image. Si l'on connait également la position du bout du nez on peut facilement définir un repère associé au visage et estimer les différents angles de rotation.



(a) (b)

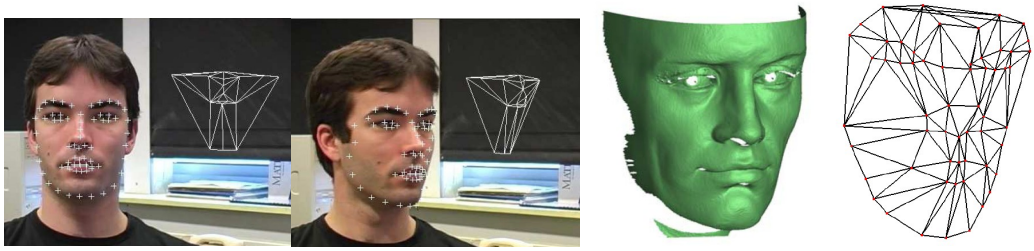
FIGURE 6.1 – Approche géométrique proposée par Gee et Cipolla [1994]. (a) Contraintes géométriques définies *a priori*.(b) Repère associé au visage défini à partir de 5 points identifiés dans l'image.

[Nikolaidis et Pitas, 2000] utilisent également les propriétés de symétrie du visage : lorsque le plan du visage est parallèle au plan image, le centre des yeux et de la bouche forment un triangle isocèle et les déformations de ce triangle permettent d'estimer les rotations hors plan.

La principale faiblesse de ces méthodes est que l'estimation de l'orientation repose sur

un faible nombre de points. Ainsi, si l'erreur dans l'extraction des points image est grande, la pose qui en résulte sera de mauvaise qualité.

Pour augmenter la robustesse de l'algorithme, il est donc préférable d'augmenter le nombre de points image. Ces méthodes combinent souvent un modèle de forme 3D et une méthode de détermination de la pose. Cette dernière s'appuie sur des points caractéristiques identifiés sur le modèle et leurs correspondants dans l'image. La solution la plus simple consiste à utiliser un modèle rigide que l'on adapte à des points du visage, qui se ne se déplacent que peu ou pas sous l'effet des expressions faciales (le nasion par exemple). Il n'existe toutefois pas un modèle générique qui convienne à toutes les morphologies. Martins et Batista [2008] utilisent un modèle de visage spécifique créé à partir d'un scan 3D du visage de la personne. Lorsque l'on ne dispose pas d'un tel dispositif, Mercier [2007] propose de générer un modèle 3D rigide adapté à une personne à l'aide d'une séquence d'images labélisées de la personne et d'une méthode de type *Structure from Motion* (SfM). Lorsque la géométrie de l'objet est connue, l'algorithme d'estimation de pose POSIT a souvent été employé pour déterminer la pose d'un modèle 3D rigide.



(a) (b)

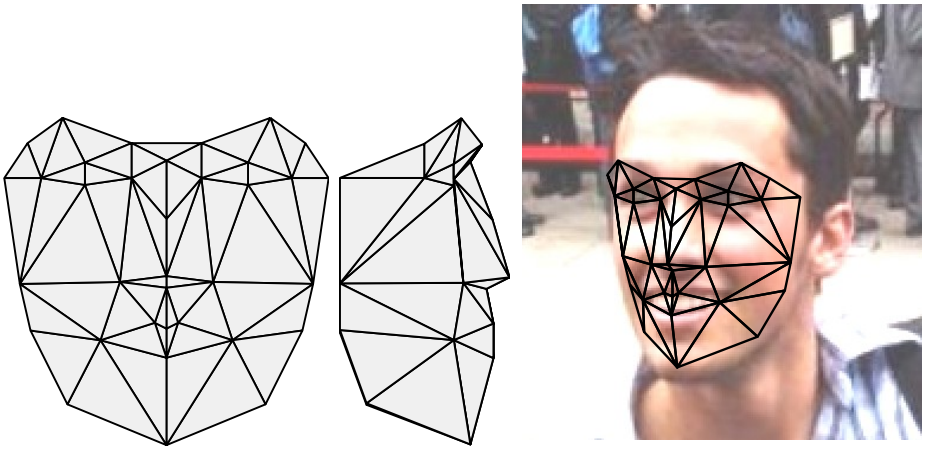
FIGURE 6.2 – Exemples de modèles rigides : (a) à partir d'une méthode de SfM [Mercier, 2007] et (b) à partir d'un scan 3D [Martins et Batista, 2008].

Les méthodes d'estimation de la pose par un modèle déformable que nous avons déjà présentées à la page 43, section 2.2 de ce document sont particulièrement bien adaptées à ce type de problèmes. Elles permettent à la fois d'utiliser tous les points identifiés lors de la phase d'alignement, y compris ceux présents sur des zones très déformables telles que la bouche, et de s'adapter aux différentes morphologies. On espère ainsi augmenter la robustesse (en augmentant le nombre de points utilisés dans l'estimation) et la précision (en diminuant le biais du modèle) de l'estimation de la pose.

6.1.3 Choix de la méthode

Nous avons choisi d'utiliser notre méthode récursive d'estimation de la pose et de la forme d'un modèle analytique 3D (*cf.* section 2.2.2 page 45). Ainsi nous pourrons exploiter tous les points du visage identifiés durant la phase d'alignement. De plus, cette méthode ne nécessite pas de données d'apprentissage car elle utilise un modèle 3D analytique. Nous ne pouvons toutefois pas utiliser directement le modèle analytique Candide comme nous le faisions précédemment, car les points du modèle n'ont pas toujours leur homologue sé-

mantique dans le modèle de forme 2D utilisé lors de la phase d'alignement et inversement. Nous sommes donc parti du modèle Candide et nous avons manuellement supprimé, déplacé et redéfini certains de ces points afin d'obtenir une bonne correspondance entre les deux modèles. Nous avons également ajouté un vecteur de déformation qui gère les déplacements en profondeur des points situés sur la bordure du visage. La figure 6.3 illustre le modèle 3D ainsi obtenu.



(a) (b)

FIGURE 6.3 – Modèle candide modifié. (a) Modèle de face et de profil. (b) Instance du modèle dans une image.

6.2 Evaluation

Dans cette section, nous définissons notre protocole d'évaluation et nous exposons les résultats obtenus sur différentes bases de données de test.

6.2.1 Jeux de données

Les tests ont porté sur les bases que nous avons utilisées pour évaluer l'alignement : PIE, Pointing et IMM. Nous avons également ajouté 50 images de la base *Labeled Faces in the Wild* [LFW, Huang *et al.*, 2007b]. Les images ont été sélectionnées aléatoirement. Il s'agit d'images très variées et de faible résolution (la distance interoculaire est d'environ 40 pixels) issues du web. La figure 6.4 présente quelques exemples de cette base.

Il est difficile de mener une évaluation sur différentes bases car les informations de pose, quand elles existent, sont souvent peu précises et/ou introduisent un biais propre au protocole d'acquisition des données. Nous avons choisi de définir nous-mêmes la vérité terrain afin de la rendre homogène sur l'ensemble des bases. Nous sommes conscients que nous introduisons un biais dans la vérité terrain mais ce biais sera le même pour toutes

les bases. De plus, nous pourrons l'évaluer à partir d'images de synthèse dont la pose est parfaitement connue.

FIGURE 6.4 – Exemples d'images de la base LFW utilisées dans notre évaluation

6.2.2 Evaluation de la vérité terrain

Dans cette évaluation, nous allons mesurer le biais introduit dans la vérité terrain à l'aide de deux jeux de données. Le premier jeu contient les images des 30 individus de la base FacePix pour des rotations horizontales allant de -40° à 0° ou de 0° à 40° avec un pas de 5°. Au total, nous disposons de 270 images labélisées. Pour chaque image nous utilisons notre méthode d'estimation de pose pour calculer $\mathbf{R}_i$, la matrice de rotation du modèle 3D par rapport au repère de la caméra. Soit $\mathbf{R}_0$ la matrice de rotation du modèle 3D pour un visage de face. Pour chaque image on peut alors calculer la matrice $\mathbf{R}_\alpha = \mathbf{R}_0 {\mathbf{R}_i}^T$ la matrice de rotation estimée pour passer du visage de face au visage tourné. De cette matrice de rotation, on peut extraire l'axe de rotation et la valeur de l'angle de rotation du modèle. L'axe de rotation de la tête n'est pas forcément colinéaire à l'un des vecteurs du repère caméra. La direction donnée par ce vecteur est difficilement interprétable. Par contre, la valeur de l'angle de rotation estimée doit correspondre à la valeur de l'angle de rotation de la caméra qui est connue. La figure 6.5 illustre l'erreur d'estimation de cet angle en fonction

de l'amplitude de la rotation.

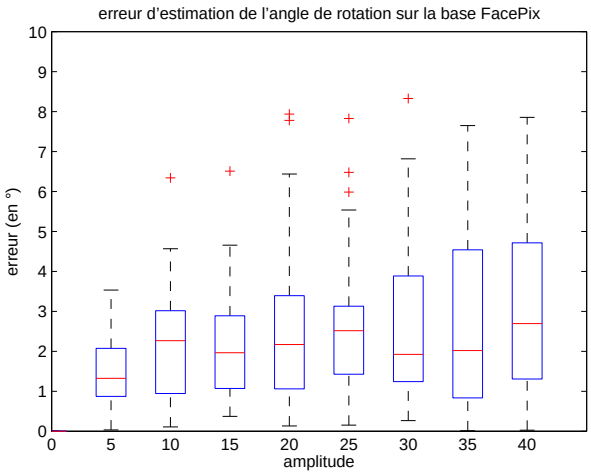


FIGURE 6.5 – Erreur d'estimation sur la base FacePix de la valeur de l'angle de rotation en fonction de l'amplitude de la rotation.

L'erreur moyenne obtenue est de 2.4° pour l'ensemble des poses. On remarque que cette erreur a tendance à augmenter à mesure que l'amplitude de la rotation augmente. Ce phénomène s'explique peut-être en partie par le fait que certains points caractéristiques du visage sont occultés lorsque le visage effectue une rotation horizontale (les points à proximité des oreilles par exemple). A la place, ce sont des points le long des limbes du visage qui sont sélectionnés dans l'image. Notre méthode, qui prend en compte ces points, aura alors tendance à légèrement sous-estimer l'angle de rotation pour les plus fortes amplitudes.

Cette base de données nous donne une indication sur l'erreur faite par notre méthode sur l'estimation de la pose. Il n'est toutefois pas possible de dissocier les erreurs liées à notre méthode de celles liées aux incertitudes de mesure de la plateforme d'acquisition. De plus, cette base permet uniquement d'évaluer les erreurs associées aux rotations horizontales du visage.

Nous avons donc décidé d'utiliser des images de synthèse pour analyser plus finement les erreurs. Nous avons généré 28 images de synthèse de visages présentant des variations de pose, d'expression et de morphologie (*cf.* figure 6.6). Comme dans l'expérience précédente on calcule pour chaque image la matrice de rotation pour passer d'un visage dans une pose de référence (visage de face, même identité, expression neutre) à la configuration du visage dans l'image considérée. Par contre cette matrice est le fruit de rotations successives suivant les axes pan, tilt et roll. On extrait les valeurs de ces angles de rotation et on les compare à celles obtenues par la vérité terrain. L'erreur moyenne d'estimation de la valeur de l'angle de rotation est de 1,7° pour pan, de 7° pour tilt et de 1,6° pour roll. On remarque que l'erreur associée à l'angle de rotation tilt est beaucoup plus importante que pour les deux autres rotations. Cette constatation s'explique probablement par le fait qu'une rotation suivant l'axe horizontal a moins d'impact sur le déplacement apparent des points qu'une rotation

suivant les autres axes. De plus, les méthodes de détermination de la rotation en pan et roll tirent profit de la symétrie verticale du visage pour obtenir une estimation plus robuste.

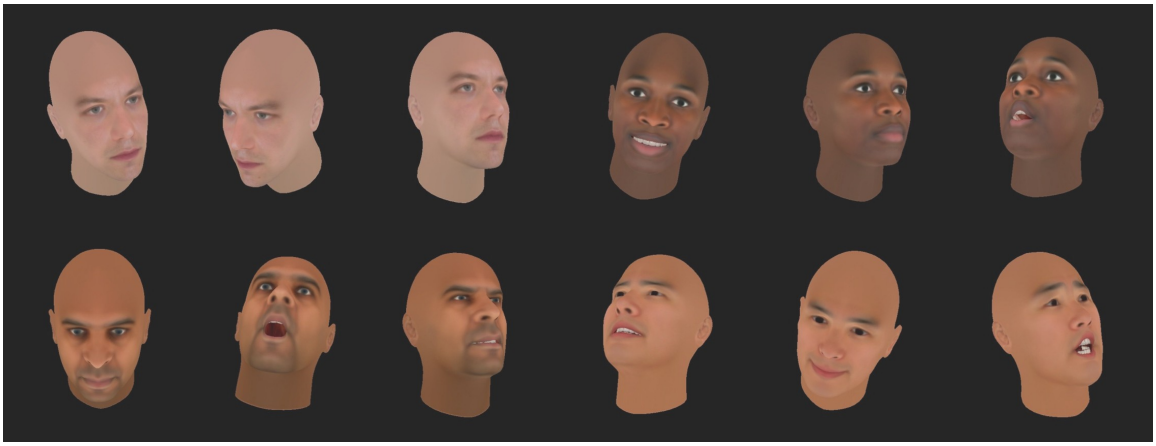


FIGURE 6.6 – Exemples d'images de synthèse utilisées

6.2.3 Evaluation de l'estimation de la pose par alignement

Nous évaluons, dans cette partie, les performances de notre méthode d'estimation de la pose par alignement. Les résultats présentés s'appuient sur une chaîne de traitements entièrement automatisée : détection du visage par la méthode de Viola et Jones [2004], détection des yeux par la méthode de Milgram *et al.* [2005], alignement du modèle 2D par BiBAM et enfin, estimation de la pose de la tête à l'aide du modèle 3D analytique [Bailly et Milgram, 2008b]. Nous présenterons, dans un premier temps, les résultats de l'alignement obtenus sur la base LFW, car elle n'a pas été évaluée au chapitre précédent. Nous exposerons ensuite les résultats de l'estimation de la pose, obtenus sur chaque base de test.

6.2.3.1 Résultats de l'alignement pour les images de la base LFW

Nous appliquons le même protocole expérimental que pour les autres bases de données (*cf.* section 5.5.8 page 129). La figure 6.7 illustre les résultats obtenus avec une initialisation automatique des yeux. La méthode BiBAM améliore substantiellement l'alignement initial : 64% des modèles alignés ont une erreur inférieure à 15% de la distance interoculaire contre à peine 12% des modèles à l'initialisation. Les résultats finaux de l'alignement sont moins bons que ceux obtenus sur la base IMM (plus de 85% ont une erreur inférieure à 15% de la distance interoculaire) mais LFW présente une plus grande variété d'images et le modèle est initialisé plus loin (12% des modèles à l'initialisation ont une erreur inférieure à 15% sur la base LFW contre 65% pour la base IMM).

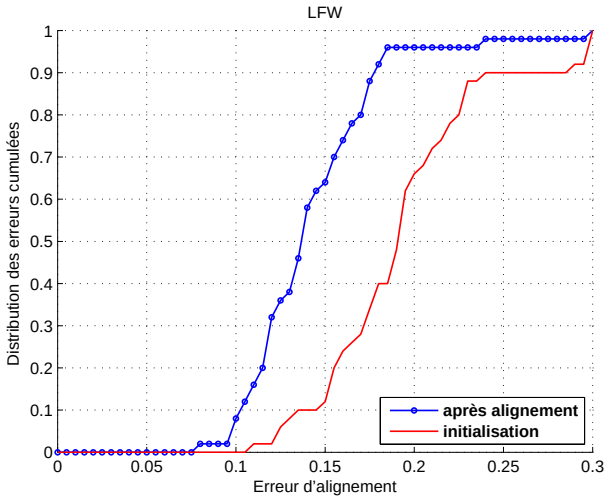


FIGURE 6.7 – Distribution des erreurs d'alignement sur 50 images de la base LFW

6.2.3.2 Résultats de l'estimation de la pose

Le tableau 6.1 regroupe les résultats obtenus sur les différentes bases de test. On remarque que l'erreur associée à l'estimation de la rotation suivant l'axe vertical est moins importante que celle suivant l'axe horizontal. Ceci confirme nos observations précédentes et est en adéquation avec les résultats obtenus par d'autres méthodes [Murphy-Chutorian et Trivedi, 2009].

Les plus mauvais résultats obtenus pour l'estimation de la rotation suivant pan est de 11° pour la base Pointing et sont liés aux problèmes d'alignement que nous avons soulignés au chapitre précédent. Pour les autres bases, l'erreur varie entre 4° pour la base PIE et 9° pour la base LFW, supposée très difficile.

L'erreur d'estimation de la valeur de l'angle roll est faible (inférieure à 3° pour les quatre bases). La précision des résultats s'explique certainement par la bonne qualité de l'alignement du modèle au niveau des yeux et de la bouche. De plus, pour de faibles rotations en *pan* (inférieures à 20°), une rotation suivant roll correspond à une rotation dans un plan parallèle au plan image.

La figure 6.8 présente des exemples d'alignement du modèle sur des images de la base LFW.

Tout comme dans le chapitre 3, nous avons évalué notre méthode par alignement sur la base Pointing 04. On ne peut toutefois pas comparer directement les résultats obtenus par nos deux méthodes, globale et par alignement, pour plusieurs raisons. Tout d'abord, l'ensemble des images considérées n'est pas le même dans les deux évaluations. En effet, nous ne sélectionnons que les visages détectés par l'algorithme de Viola Jones pour évaluer la méthode par alignement. Ces images correspondent à des variations de ±30° en pan et tilt. De plus, l'apprentissage de la méthode globale du chapitre 3 utilise la première série

FIGURE 6.8 – Résultats obtenus sur quelques images de la base LFW. Le modèle en rouge correspond au modèle à l'initialisation et le modèle en noir, au résultats final de l'alignement. En-dessous de chaque image est reportée l'erreur d'alignement en % (Ali) du modèle à l'initialisation et à la fin de l'alignement ainsi que l'erreur d'estimation en pan (P), et tilt (T) et en roll (R) en degrès

TABLE 6.1 – Erreur moyenne et médiane en degrés sur les différentes bases de test

	Erreur pan		Erreur tilt		Erreur roll	
	Moyenne	Médiane	Moyenne	Médiane	Moyenne	Médiane
PIE (Groupe 3)	3.9°	2.6°	6.8°	4.4°	1.4°	1.1°
Pointing 04 (Groupe 4)	10.9°	8.2°	7.3°	5.3°	2.8°	1.5°
IMM (Groupe 5)	6.1°	4.1°	5.7°	4.6°	1.6°	1.1°
LFW	8.6°	7.7°	10.1°	9.1°	2.5°	1.8°

d'images de la base Pointing, contrairement à l'approche par alignement qui n'utilise aucune image de cette base lors de l'apprentissage.

6.3 Conclusion

Ce chapitre a montré la viabilité d'une approche par alignement d'un modèle de forme 2D pour l'estimation de la pose de la tête. Cette approche a été validée sur quatre bases de données, notamment sur 50 images de la base LFW. Cette base est particulièrement difficile car elle regroupe des images acquises dans des environnements très divers et non-contraints. Nous avons également soulevé le problème de la non-homogénéité des bases de données pour l'évaluation de la pose. Nous avons alors décidé d'utiliser la position des points caractéristiques du visage, donnée par la vérité terrain pour aligner un modèle déformable 3D et extraire l'information de pose. Cette approche a été validée sur des images de synthèse dont la pose est parfaitement connue. La dernière observation de ce chapitre concerne la précision attachée à l'estimation de chaque angle de rotation ; l'erreur d'estimation de valeur des angles pan et roll est nettement plus faible que l'erreur associée à l'angle tilt.

Bilan et perspectives

Conclusion générale

Cette thèse a abordé le problème de l'estimation de la pose de la tête dans des images fixes monoculaires. Nous avons proposé dans un premier temps, une méthode par comparaison à des images de synthèse. A partir d'une image de visage labélisée, on estime les paramètres de déformation du modèle géométrique Candide par un processus itératif de type EM. On génère un ensemble d'images de synthèse pour différentes orientations et expressions faciales. Pour déterminer la pose, on compare l'image à analyser à chaque vue de synthèse à l'aide d'une mesure de ressemblance entre contours orientés, proche de la distance de Hausdorff. Le principal avantage de cette méthode est qu'elle ne nécessite ni base de données, ni phase d'apprentissage. Toutefois, cette approche nous a montré des limites importantes en terme de précision, de robustesse et de temps de calcul.

Nous nous sommes alors tourné vers une approche globale par apprentissage supervisé pour déterminer la relation entre l'apparence d'un visage et sa pose. Dans ce contexte, nous avons proposé BISAR, une nouvelle méthode de sélection de descripteurs adaptée aux problèmes de régression. Elle consiste à sélectionner itérativement les entrées d'un réseau de neurones. Chaque entrée est associée à un descripteur sélectionné à l'aide de notre critère fonctionnel flou, FFC. Ce dernier mesure sur la base d'apprentissage, la dépendance fonctionnelle de chaque descripteur avec les valeurs que l'on souhaite prédire, *i.e.* les angles de rotation du visage. Pour que les informations apportées par les descripteurs soient complémentaires, le processus de sélection intègre une stratégie de *boosting* qui modifie l'importance de chaque exemple au fil des itérations. Nous avons comparé BISAR avec un Réseau de Neurones à Convolution (CNN) sur deux bases de données. Le CNN et BISAR ont présenté des résultats équivalents sur la base FacePix, mais BISAR a surpassé le CNN sur la base Pointing 04. Nous avons obtenu des résultats du niveau de la meilleure méthode de l'évaluation CLEAR 07 sur cette même base de données.

Nous nous sommes également intéressé à l'estimation de la pose par alignement d'un modèle déformable. Cette approche présente deux avantages par rapport aux méthodes globales. Elle est à la fois moins sensible à la qualité de la localisation du visage dans l'image et est réputée plus précise dans l'estimation de la pose. Le processus d'alignement repose sur une fonction de coût qui évalue la qualité de l'alignement. Cette fonction est apprise par BISAR à partir d'exemples de modèles plus ou moins bien alignés. Les évaluations – effectuées sur différentes bases présentant des variations de pose, d'identité, d'illumination et de condition de prise de vues – montrent que notre méthode d'alignement de visages donne de meilleurs résultats que la méthode *Boosted Ranking Model*. On extrait alors l'information de pose par une mise en correspondance des points image issus de la phase d'alignement avec les points d'un modèle déformable 3D analytique.

Perspectives

Nous proposons ci-après, quatre axes de recherche dans la continuité des travaux présentés dans cette thèse.

Estimation de la direction du regard. Dans le projet PILE, cadre applicatif à l'origine de cette thèse, nous avions à concevoir un module de détermination de la direction du regard, dont l'estimation de l'orientation de la tête constitue le premier étage. La prochaine étape concerne l'estimation de la direction du regard par rapport au repère associé au visage. Pour y parvenir, plusieurs solutions peuvent être envisagées. On peut, par exemple, localiser finement les pupilles et les commissures des yeux et déterminer la direction du regard à l'aide d'un modèle géométrique de l'œil [Ishikawa *et al.*, 2004]. Les points caractéristiques des yeux peuvent être extraits de manière robuste par un modèle d'apparence spécifique de l'œil [Orozco *et al.*, 2009]. On peut également imaginer une approche globale qui estime la direction du regard en fonction de l'apparence des yeux. Baluja et Pomerleau [1993] apprennent cette relation à l'aide d'un réseau de neurones.

Exploitation des données PILE. Dans une optique de comparaison avec des méthodes existantes, nous avons préféré exploiter les images issues de bases de données publiques. La morphologie d'un visage de bébé est différente de celle d'un adulte. L'adaptation des méthodes présentées dans cette thèse implique, par conséquent, un réapprentissage spécifique à partir d'images du projet PILE (ou tout du moins, d'images de visages de bébés). Par ailleurs, nous avons fait le choix d'une solution monoculaire pour l'estimation de la pose de la tête afin qu'elle soit :

- Générique et puisse fonctionner avec le plus grand nombre de périphériques d'acquisition vidéo.
- Peu dépendante de l'étape de calibration du réseau de caméras.

En revanche, nous pourrons exploiter les données issues des différentes caméras du projet PILE pour rendre l'estimation plus robuste (en choisissant, par exemple, la vue qui n'est pas occultée ou en fusionnant les estimations de chaque caméra). Le réseau sera également mis à contribution pour estimer le contact visuel mère-bébé car le visage de la mère et du bébé ne sont pas couvert par le même champ de vision.

Combinaison de l'approche globale et par alignement. L'estimation de la pose du visage par un modèle déformable 3D est précise lorsque les points caractéristiques du visage sont correctement localisés. Toutefois, notre méthode d'alignement BiBAM, sur laquelle repose la localisation de ces points, échoue lorsque le modèle est initialisé trop loin de la position à atteindre. De plus, il n'est pas possible de gérer toutes les orientations du visage avec un seul modèle 2D de forme et d'apparence. On pourrait alors faire appel à notre méthode globale pour choisir le modèle à adapter (face, 3/4 ou profil par exemple) et l'initialiser grossièrement dans l'image, puis affiner l'estimation de la pose avec BiBAM. Une autre

manière de combiner nos deux approches serait d'utiliser leurs estimations pour définir une mesure de confiance : cette dernière doit être d'autant plus forte, que les estimations de ces deux méthodes sont semblables.

Vers un modèle 3D. Il serait intéressant de substituer un modèle de forme 3D à l'actuel modèle 2D. On aurait ainsi directement accès à l'information de pose à la fin du processus d'alignement. Le modèle pourrait, de plus, couvrir une vaste amplitude de poses, sans avoir recours à plusieurs modèles. Le passage à un modèle 3D n'est toutefois pas trivial et soulève d'autres problématiques. Il faudra considérer, par exemple, les auto-occultations provoquées par les fortes rotations hors-plan de la tête et gérer l'absence de texture dans ces parties cachées du modèle.

Les travaux de recherche présentés dans cette thèse trouvent également des prolongements dans les projets actuels et futurs du laboratoire. Dans le cadre de sa collaboration industrielle avec la société Majority Report, l'ISIR développe une chaîne d'outils de mesures fines du comportement (pose de la tête, direction du regard) de personnes placées en situation d'achat. Le projet ANR IMMEMO (IMMersion 3D basée sur l'interaction EMOtionnelle) vise à développer un système d'immersion 3D dédié à la formation interactive. Il s'agit de capturer et reconnaître les expressions du visage de la personne immergée (l'apprenant) afin de faciliter son interaction avec un agent conversationnel autonome dont le comportement s'adaptera de façon semi-automatique à celui de l'apprenant. L'équipe Perception Artificielle doit donc concevoir un système de MoCap (Motion Capture) facial sans marqueur et en situation dynamique, robuste vis à vis des diverses sources de variabilités (pose, émotions) pouvant apparaître dans les séquences vidéos. Le laboratoire a également répondu à l'appel à projet (ANR ContInt 2010) Défi REPERE (Reconnaissance de PERsonnes dans des Emissions audiovisuelles) dont l'objectif est de développer un système non contraint de reconnaissance multimodale de personnes dans des émissions audiovisuelles

Annexes

Annexe A

Localisation de la tête

Dans la plupart des expériences présentées dans ce manuscrit, nous utilisons la version OpenCV [1] du détecteur de Viola et Jones [2001] (amélioré par Lienhart et Maydt [2002]) pour localiser le visage dans une image. Les résultats d'apprentissage disponibles dans cette librairie permettent de détecter des visages de face et de profil.

Toutefois, il n'existe pas, à notre connaissance, de détecteurs publics de visages multi-orientations capables de couvrir l'ensemble des poses de la base Pointing 04. Pour nous comparer aux autres méthodes de l'évaluation CLEAR 07, nous avons dû concevoir une méthode de localisation de visage dont le principe est illustré dans la figure A.1. Bien que développée spécifiquement pour l'évaluation de la base Pointing, cette méthode peut s'appliquer à d'autres bases de données d'images couleur de visages.

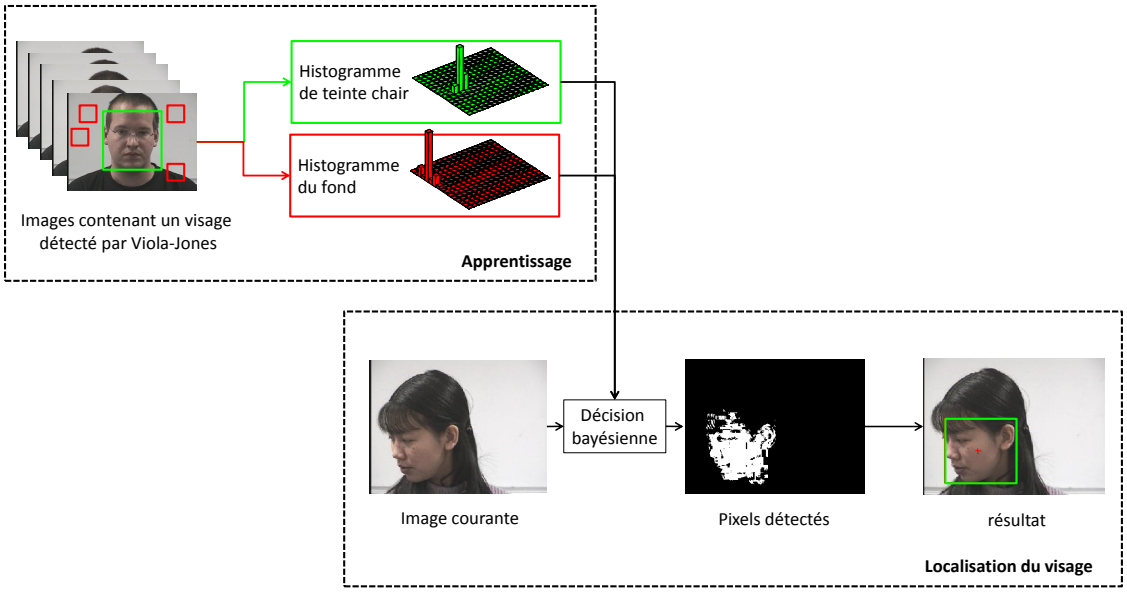


FIGURE A.1 – Principe de notre méthode de localisation du visage

Elle s'appuie sur la détection de la teinte chair. Dans un premier temps, les images de face sont détectées à l'aide du détecteur de Viola-Jones. Les pixels contenus à l'intérieur de

1. OpenCV est une librairie open-source de fonctions pour la vision par ordinateur disponible à l'adresse http ://sourceforge.net/projects/opencvlibrary/

la zone détectée sont utilisés pour construire un histogramme à deux dimensions à partir des canaux de teinte et de saturation de l'espace colorimétrique HSV. L' histogramme est normalisé et chacune de ses classes correspond à la probabilité pour un pixel d'appartenir au visage (teinte chair). D'autres pixels sont séléctionnés aléatoirement en dehors de la zone du visage pour construire l'histogramme du fond. Dans une nouvelle image de la base, on détecte alors les pixels du visage par une décision bayesienne qui repose sur les histogrammes de teinte chair et non-chair. La taille de la boite englobante du visage est proportionnelle à l'écart type des pixels suivant l'axe horizontal et l'axe vertical de l'image. Nous avons choisi deux ecart type dans nos expérience.

Dans cette méthode, seuls les visages de la base d'apprentissage sont utilisés et aucune étape d'étiquetage manuel n'est requise. Le principal inconvénient de cette méthode concerne la précision. Dans certains cas, par exemple, les pixels du cou seront détectés comme appartenant au visage et modifierons les limites de la boîte englobante.

Annexe B

Système d'acquisition multicamera

Cette annexe présente le réseau de caméras mis en œvre dans le cadre du projet PILE. Il s'agit d'un extrait de l'article [Bailly *et al.*, 2006b].

B.1 Choix technique

Pour analyser les gestes et les regards du bébé, nous avons opté pour un système orienté vision plutôt que d'autres capteurs tels que l'on peut voir dans [Mathieu et Fuchs, 2006]. Ce choix a été motivé par trois raisons :

- l'analyse d'image permet de répondre aux objectifs que nous avons définis précédemment ;
- l'équipe des cliniciens utilise les enregistrements vidéo pour leurs observations. Ainsi nous simplifions le mode opératoire et la mise en correpondance des différentes données ; nous réduisons également le nombre de capteurs ;
- le système est non intrusif et préserve la spontanéité de l'interaction mère-bébé.

B.2 Description de l'installation

Nous avons été amené à définir un réseau de cameras (figure B.1).

Trois champs de vision, composés d'au moins deux caméras pour extraire la géométrie 3D des images, sont nécessaires. Un premier (caméras 1 et 8, en bleu sur la figure B.1) est restreint au visage du bébé pour estimer la direction de son regard (cela requière en effet une forte résolution). Un second (caméras 2,4,6 et 7 en jaune) permet la capture de ses gestes. Ce champ se compose de quatre caméras afin de limiter l'influence des nombreuses occlusions qui interviennent au cours du tournage. Le troisième (caméras 3 et 5 en rouge) est dédié à la localisation de la mère. Les caméras sont des caméscopes miniDV reliés à des enregistreurs numériques. Elles sont synchronisées à l'aide de signaux lumineux visibles par toutes les caméras. L'extraction de l'information 3D des différents flux vidéo nécessite une étape préalable de calibrage. Nous présentons notre méthode dans la section suivante.

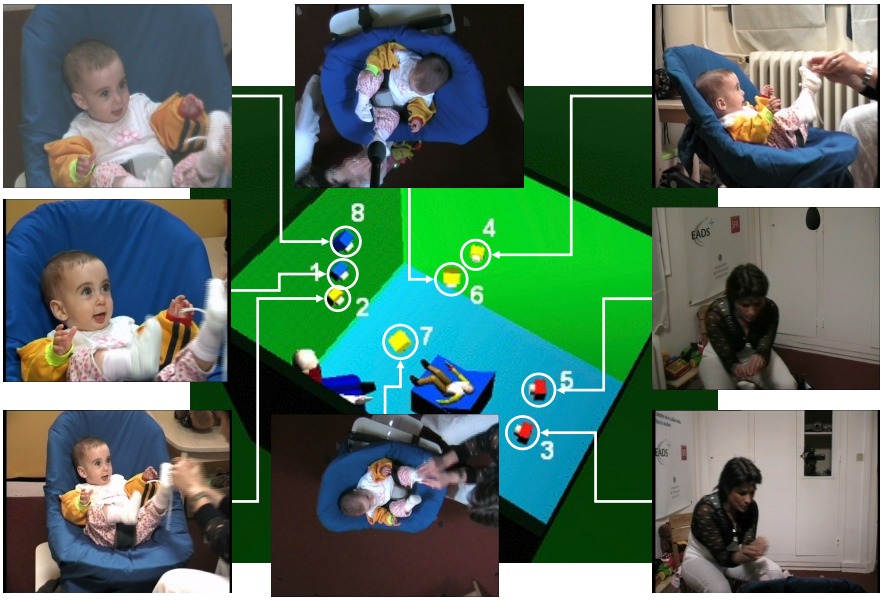


FIGURE B.1 – Emplacement des caméras

B.3 Calibrage des caméras

Le calibrage des caméras est une étape d'estimation de deux types de paramètres :

- les paramètres intrinsèques : ce sont les paramètres propres à une caméra tels que la longueur focale. Ils sont regroupés dans une matrice K et définissent le passage du repère caméra au repère image (cf. figure B.2) ;
- les paramètres extrinsèques : ils correspondent à la position et l'orientation des caméras dans la scène. Ils définissent donc le passage du repère monde au repère caméra (combinaison d'une rotation R suivant les 3 axes et d'une translation t)

Il existe de nombreuses méthodes de calibrage que l'on peut répartir en deux catégories. Les méthodes qui s'appuient sur des informations métriques connues a priori et les méthodes d'autocalibrage [Quan et Triggs, 2000]. Elles limitent les manipulations préalables (pas d'enregistrement d'une mire de calibrage par exemple) mais elles sont moins robustes et moins précises. Ayant la possibilité d'agir sur la scène avant chaque enregistrement, nous nous sommes tournés vers la première catégorie de méthodes. Cette dernière peut se décomposer en plusieurs sous-catégories en fonction du type d'objets utilisés pour calibrer. Certaines utilisent un pointeur laser Svoboda *et al.* [2005] mais elles ne sont pas adaptées dans notre cas puisque les caméras de notre réseau ne couvrent pas toute le même champ de vision. Les calibrages par mire plane Zhang [2000] donnent de très bon résultats mais elles sont contraignantes car un opérateur doit manipuler la mire successivement devant chaque caméra. Par ailleurs, l'opérateur n'a pas de retour visuel direct de ce qu'il est en train d'enregistrer et ne peut, par conséquent, pas savoir si le cadrage est correct. Nous

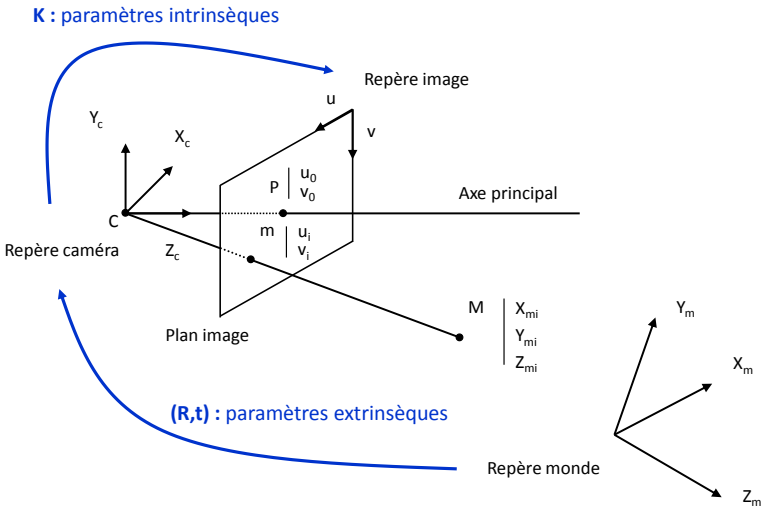


FIGURE B.2 – Modèle projectif d'une caméra : un point M de l'espace se projète en un point m sur le plan image.

avons donc privilégié l'emploi d'une mire 3D fixe. L'opérateur peut alors poser la mire dans la scène et vérifier qu'elle est bien positionnée.

B.3.1 Description de la méthode

Notre méthode propose deux originalités. D'une part, nous avons conçu une mire de calibrage tridimensionnelle adaptée à notre réseau de caméras (figure B.3). Elle a été pensée de manière à ce que chaque caméra puisse observer au moins deux faces non coplanaires et que les paires stéréoscopiques (caméras 3 et 5 par exemple) observent les mêmes plans. D'autre part, les caméras sont déplacées entre chaque tournage. Il est donc nécessaire de les recalibrer à chaque fois. Dans un souci d'automatisation de cette étape, nous avons alors conçu un algorithme de calibrage fondé sur une segmentation couleur de la mire. Il s'effectue en 3 phases distinctes :

- La localisation des points d'intérêt de la mire. Une segmentation par la couleur permet d'extraire les informations de position des amers de la mire. Elle repose sur un apprentissage préalable des cinq couleurs de la mire.
- Le calcul des paramètres internes des caméras à partir de l'estimation de la matrice de projection.
- Le calcul des paramètres externes : on détermine pour chaque caméra une matrice de passage vers un repère commun (le repère d'une caméra par exemple) en s'appuyant sur les plans de la mire que les caméras ont en commun.

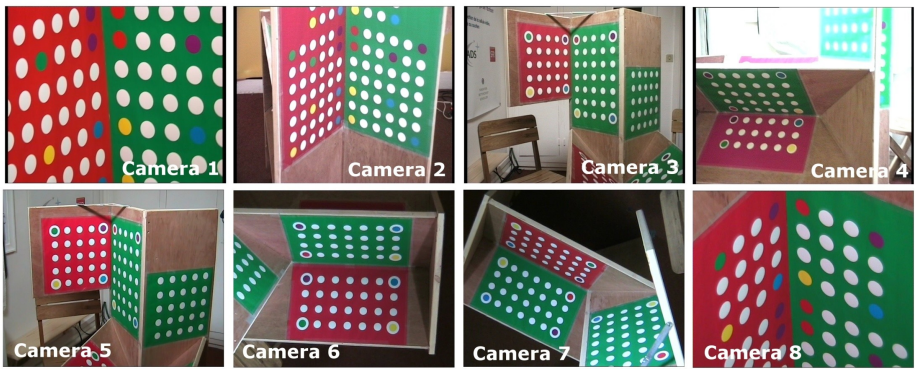


FIGURE B.3 – mire vue par les 8 caméras

B.3.2 Localisation des points d'intérêt de la mire

Segmentation couleur : Nous effectuons dans un premier temps un apprentissage pour chaque couleur de la mire. Dans une séquence vidéo, nous sélectionnons une zone de la couleur à apprendre. Chaque pixel de cet échantillon représente un point dans l'espace RGB que nous projetons sur les trois plans rouge-vert rouge-bleu et vert-bleu. Ainsi nous obtenons un ensemble de points caractéristiques d'une couleur. La figure B.4 présente un exemple appliqué à la couleur mauve. En renouvelant cette opération sur une séquence de quelques secondes, on modélise implicitement l'instabilité statique du capteur pour un éclairage donné.

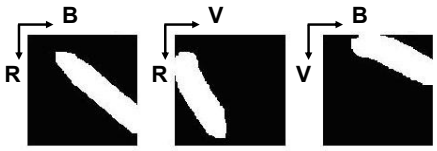


FIGURE B.4 – les zones blanches correspondent à l'ensemble des pixels caractéristiques de la couleur mauve dans les plans bleu-rouge, vert-rouge et bleu-vert

Pendant la phase de traitement, nous pouvons alors très simplement rechercher les cinq couleurs caractéristiques de la mire. Pour une couleur donnée et pour chaque pixel de l'image, nous testons son appartenance à l'ensemble des pixels caractéristiques de la couleur (la figure B.5 montre un exemple de segmentation couleur).

La méthode que nous proposons présente trois intérêts :

- Elle est très simple à implémenter ;
- Les traitements sont très rapides et nécessitent peu de ressources ;
- Elle est adaptée au capteur et aux conditions d'éclairage utilisés pendant l'apprentissage.

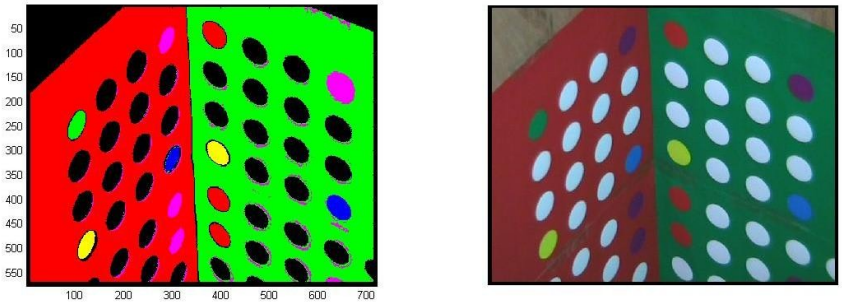


FIGURE B.5 – L'image de gauche présente le résultat de la segmentation couleur. Les couleurs affichées dans l'image de gauche correspondent aux cinq couleurs caractéristiques de la mire détectées à partir de l'image de droite

Localisation des points : Nous pouvons alors déterminer de manière automatique, la position des amers de couleurs de la mire (que nous appellerons coins de la mire par la suite) pour chaque vue puisqu'ils sont définis de manière unique dans l'image.

Nous calculons alors l'homographie entre la position des coins dans le repère mire et la position des coins dans le repère image. C'est la même transformation qui s'applique à tous les amers de la mire. On trouve ainsi les positions approximatives des centres des amers dans l'image. On affine la localisation en calculant le centre de gravité de tous les amers identifiés. On obtient ainsi la position subpixelique des amers dans le repère image (figure B.6).

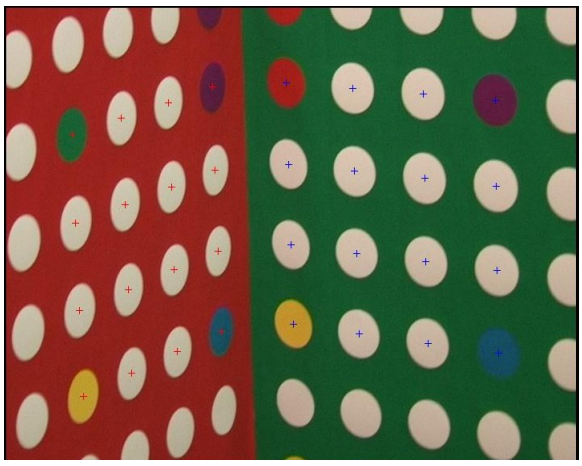


FIGURE B.6 – Détection automatique des disques blancs de la mire

B.3.3 Estimation des paramètres intrinsèques

Nous avons à présent un ensemble de correspondance entre des points M_i du repère monde associé à la mire et leur projeté m_i dans le repère image. Cette relation est donnée par l'équation B.1 :

$$m_i = PM_i \tag{B.1}$$

Avec P la matrice de projection. Elle est estimée grâce à l'algorithme de Transformation Linéaire Directe Normalisée décrite dans Hartley et Zisserman [2003]. L'erreur de reprojection des points de la mire est comprise entre 0.3 et 0.9 pixels en fonction des caméras. La matrice de projection se décompose de la manière suivante :

$$P = K[R|t] \tag{B.2}$$

avec K une matrice triangulaire supérieure de la forme :

$$K = \begin{pmatrix} \alpha_u & s & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{B.3}$$

avec :

- α_u : le facteur d'échelle suivant l'axe u ;
- α_v : le facteur d'échelle suivant l'axe v ;
- $(u_0; v_0)^T$: les coordonnées du point principal ;
- s : le facteur de "verticalité" du plan de projection ;

On extrait la matrice K en utilisant la décomposition RQ

B.3.4 Paramètres extrinsèques des caméras

Les paramètres extrinsèques des caméras sont également obtenus à partir de la matrice de projection Hartley et Zisserman [2003]. Ces paramètres donnent les positions et les orientations des caméras par rapport à un repère associé aux plans vus par les caméras. Le repère monde n'est donc pas le même pour toutes les caméras puisque elles n'observent pas toutes les mêmes paires de plans. Nous devons donc exprimer les matrices de passage des repères mondes propres à chaque caméra à un repère monde commun (nous avons choisi de nous placer dans le repère de la caméra 3).

Les caméras 1,2,3,5 et 8 regardent des mires plaquées de chaque cotés de supports plans assemblés avec des angles à 90° et dont l'épaisseur est connue. Nous pouvons alors définir simplement les matrices de transformation. En effet, ce sont des composées de translations et de rotations de 180°. Nous appellerons ce nouveau repère, repère mire 1 (rm_1). Soit $T_{rm_i rc_j}$ la matrice de transformation euclidienne permettant le passage du repère mire i (rm_i) au repère caméra j (rc_j). Connaissant $T_{rm_1 rc_i}$ pour $i = \{1, 2, 3, 5, 8\}$= on a :

$$T_{rc_i rc_3} = T_{rm_1 rc_3} * T^{-1}_{rm_1 rc_i} \tag{B.4}$$

La position des caméras 6 et 7 est connue par rapport à un repère que nous nommerons le repère mire 2 (rm_2). Nous ne pouvons pas mesurer avec précision la transformation qui permet de passer du repère monde 1 au repère monde 2. Pour calculer la matrice de passage du repère caméra 7 au repère caméra 3, nous nous appuierons sur un plan observé par les deux caméras, que nous appellerons par la suite repère plan 1 (rp_1) et que l'on peut observer sur la figure B.7(a).

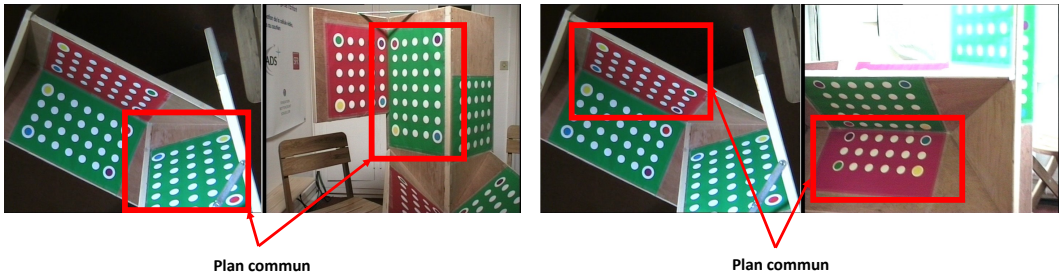
Connaissant les paramètres intrinsèques des caméras et les distances entre les points de la mire, on calcule la matrice de passage $T_{rp_1 rc_i}$ du repère plan 1, au repère caméra i (avec $i = \{3, 7\}$) par la méthode de Sturm [2000]. Ainsi on calcule la matrice de passage du repère caméra 7 au repère camera 3 :

$$T_{rc_7 rc_3} = T_{rp_1 rc_3} * T^{-1}_{rp_1 rc_7} \tag{B.5}$$

Et la matrice de passage du repère de la caméra 6 au repère de la caméra 3 en passant par le repère caméra 7 :

$$T_{rc_6 rc_3} = T_{rc_7 rc_3} * T_{rm_2 rc_7} * T^{-1}_{rm_2 rc_6} \tag{B.6}$$

Les caméras 4 et 7 observent deux parties distinctes d'un même plan (cf. figure B.7(b)) dont on peut exprimer la position des points de ce plan dans un repère commun (le repère plan 2).



(a) Plan commun aux caméras 3 et 7

(b) Plan commun aux caméras 4 et 7

FIGURE B.7 – Plans communs utilisés pour le calcul des matrices de passage entre caméras

De la même manière que pour la caméra 6, on détermine la matrice de passage du repère caméra 4 au repère caméra 7 et on en déduit la matrice de passage du repère caméra 4 au repère caméra 3.

$$T_{rc_4 rc_3} = T_{rc_7 rc_3} * T_{rc_4 rc_7} \tag{B.7}$$

$$T_{rc_4 rc_3} = T_{rc_7 rc_3} * T_{rp_2 rc_7} * T_{rp_2 rc_4} \tag{B.8}$$

Notre réseau de caméras est donc entièrement calibré et nous pouvons exprimer par triangulation, la position de n'importe quel objet de la scène observé par au moins deux caméras.

Annexe C

Publications

Revue internationale avec comité de lecture

[1] **K. Bailly** et M. Milgram, *Boosting Feature Selection for Neural Network based Regression*, **Neural Networks 22** (5-6) : 748-756, 2009.

Conférences internationales avec actes et comité de lecture

[2] T. Senechal, **K. Bailly** et L. Prevost *Automatic Facial Action Detection Using Histogram Variation between Emotional States*, International Conference on Pattern Recognition (**ICPR'10**). Istanbul, Turkey, 2010.

[3] **K. Bailly**, M. Milgram et Philippe Phothisane, *Head Pose Estimation by a Stepwise Nonlinear Regression*, International Conference on Computer Analysis of Images and Patterns (**CAIP'09**). Münster, Germany, 2009.

[4] **K. Bailly** et M. Milgram, *BISAR : Boosted Input Selection Algorithm for Regression*, International Joint Conference on Neural Networks (**IJCNN'09**). Atlanta, USA, 2009.

[5] **K. Bailly** et M. Milgram, *Head Pan Angle Estimation by a Nonlinear Regression on Selected Features*, International Conference on Image Processing (**ICIP'09**). Cairo, Egypt, 2009

[6] **K. Bailly** et M. Milgram, *Head Pose Determination using Synthetic Images*, Advanced Concepts for Intelligent Vision Systems (**ACIVS 2008**), Juan-les-Pins, France, 2008.

[7] **K. Bailly** et M. Milgram, *Recursive Shape and Pose Determination using Deformable Model*, 13th Iberoamerican Congress on Pattern Recognition (**CIARP 2008**), Havane, Cuba, 2008.

[8] **K. Bailly**, *A multi-camera system for baby gaze and gesture analysis*, IEEE International Conference on Research , Innovation and Vision for the Future (**RIVF 2007**), Doctoral Symposium, Hanoi, Vietnam, 2007.

Revue spécialisée avec comité de lecture

[9] **K. Bailly**, J. Kiss, V. Desjardins et B. Golse, *Les Productions vocales du bébé : hyperfréquences et processus d'attachement*, **Le Carnet Psy**, pp 34–37, 2005.

Communications nationales avec actes

[10] **K. Bailly**, X. Clady, R. Benosman et M. Milgram, *Système multivues pour l'analyse des gestes et du regard du bébé : application à la détection des troubles du langage*, MAnifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (**MajecSTIC 06**), Lorient, France, 2006.

Communications internationales sans actes

[11] **K. Bailly**, X. Clady, R. Benosman et M. Milgram, *Hand tracking and gaze estimation, World Congress of WAIMH (World Association for Infant Mental Health)*, Paris, France, 2006. [1]

[12] **K. Bailly**, X. Clady, R. Benosman et M. Milgram, *Visual perception for hands and gaze tracking in 3-9 months old childs*, European Congress of WAIMH, Rome, Italie, 2005.

Séminaires

[13] **K. Bailly** et M. Milgram, *Alignement d'un Modèle Facial par apprentissage de la fonction de coût*, Journée du groupe FAST Paris, France, 2009.

[14] **K. Bailly** et M. Milgram, *Détermination de la pose du visage à partir d'images de synthèse*, Journée Geste et Visage de l'action spécifique Geste et Action du GdR ISIS Paris, France, 2008.

[15] **K. Bailly**, X. Clady, R. Benosman et M. Milgram, *Système d'acquisition multicaméras dédié à l'analyse des gestes et du regard du bébé.*, Journée Jeunes Chercheurs en Visage-Gest-Mouvement du GdR ISIS Paris, France, 2006.

1. Cette communication fait partie d'un ensemble de quatre communications publiées sous : Golse, B.(2006). *The Multidisciplinary Basis of the Research Program Pile (International Program for the Children's Speech)*, Infant Mental Health Journal, **27**(3), No. 474.

Bibliographie

H. Aanaes et F. Kahl : Estimation of deformable structure and motion. *In Workshop on Vision and Modelling of Dynamic Scenes (Workshop ECCV'02)*, 2002. 44

B. Abboud et F. Davoine : Bilinear factorisation for facial expression analysis and synthesis : Technologies for interactive multimedia services. *Vision, Image and Signal Processing*, 152(2):327–333, 2005. 98

C. Achard, E. Bigorgne et J. Devars : A sub-pixel and multispectral corner detector. *In International Conference on Pattern Recognition (ICPR'00)*, volume 3, page 3971, 2000. ISBN 0-7695-0750-6. 21

J. Ahlberg : Candide - a parameterized face. `http://www.icg.isy.liu.se/candide/main.html`. 45

J. Ahlberg : Candide-3 – an updated parameterized face. Rapport technique, Dept. of Electrical Engineering, Linköping University, 2001a. 3, 45, 91

J. Ahlberg : Using the active appearance algorithm for face and facial feature tracking. *In ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (RATFG-RTS'01)*, 2001b. 91

H. Akaike : A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. 61

H. Almuallim et T.G. Dietterich : Learning boolean concept in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994. 60

R. Avnimelech et N. Intrator : Boosted mixture of experts : an ensemble learning scheme. *Neural Computation*, 11(2):483–497, 1999. 62

K. Bailly : A multi-camera system for baby gaze and gesture analysis. *In IEEE International Conference on Research, Innovation and Vision for the Future (RIVF'07), Doctoral Symposium*, 2007. 10

K. Bailly, R. Benosman, X. Clady et M. Milgram : Visual perception for hands and gaze tracking in 3-9 months old childs. clinical applications of the research program pile (international research program for children's speech). *In World Association for Infant Mental Health World Congress*, 2006a. 10

K. Bailly, X. Clady, R. Benosman, A. Chetouani et M. Milgram : Système multivues pour l'analyse des gestes et du regard du bébé : application à la détection des troubles du langage. *In MAnifestation des JEunes Chercheurs en Sciences et Technologies de l'Information et de la Communication (MajecStic'06)*, 2006b. 10, 151

K. Bailly et M. Milgram : Head Pose Determination Using Synthetic Images. *In Advanced Concepts for Intelligent Vision Systems (ACIVS'08)*, pages 1071–1080, 2008a. 43

K. Bailly et M. Milgram : Recursive shape and pose determination using deformable model. *In Iberoamerican Congress on Pattern Recognition (CIARP'08)*, pages 602–609. Springer-Verlag, 2008b. 43, 44, 139

K. Bailly et M. Milgram : Bisar : Boosted input selection algorithm for regression. *In International Joint Conference on Neural Networks (IJCNN'09)*, pages 249–255, 2009a. 63

K. Bailly et M. Milgram : Boosting feature selection for neural network based regression. *Neural Networks*, 22:748–756, 2009b. 63

M. Balasubramanian et E. L. Schwartz : The isomap algorithm and topological stability. *Science*, 295(5552):7, 2002. 24

V. N. Balasubramanian, S. Krishna et S. Panchanathan : Person-independent head pose estimation using biased manifold embedding. *EURASIP Journal on Advances in Signal Processing*, 2008:1–15, 2008. 27, 28, 36

K. Balci, E. Not, M. Zancanaro et F. Pianesi : Xface open source project and smil-agent scripting language for creating and animating embodied conversational agents. *In International conference on Multimedia (MULTIMEDIA'07)*, 2007. 3, 91

D. H. Ballard : Generalizing the hough transform to detect arbitrary shapes. *Readings in computer vision : issues, problems, principles, and paradigms*, pages 714–725, 1987. 106

S. Baluja et D. Pomerleau : Non-intrusive gaze tracking using artificial neural networks. *In Working Notes : AAAI Fall Symposium Series, Machine Learning in Computer Vision : What, Why and How ?*, 1993. 144

S. Baluja, M. Sahami et H.A. Rowley : Efficient face orientation discrimination. *In International Conference on Image Processing (ICIP'04)*, pages 589–592, 2004. 31

S. Basu, I. Essa et A. Pentland : Motion regularization for model-based head tracking. *In International Conference on Pattern Recognition (ICPR'96)*, volume 3, page 611, 1996. 30

R. Battiti : Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5:537–550, 1994. 68

H. Bay, A. Ess, T. Tuytelaars et L. Van Gool : Speeded-up robust features (surf). *Computer Vision and Image Understanding*, 110(3):346–359, 2008. 21

P. A. Beardsley, A. Zisserman et D. W. Murray : Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23:235–259, 1997. 43

M. Belkin et P. Niyogi : Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15:1373–1396, 2003. 25, 28

R. Bellman : *Adaptive Control Processes : A Guided Tour*. Princeton University Press, 1961. 68

S. Belongie, J. Malik et J. Puzicha : Matching shapes. *In International Conference on Computer Vision (ICCV'01)*, 2001. 21

Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N.L. Roux et M. Ouimet : Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. *In Neural Information Processing Systems (NIPS'03)*, 2003. 26

D. J. Beymer : Face recognition under varying pose. *In Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994. 29

P. Bühlmann et T. Hothorn : Boosting algorithms : Regularization, prediction and model fitting. *Statistical Science : a review journal of the Institute of Mathematical Statistics*, 22:477–505, 2007. 77

C. Bishop : *Neural Networks for Pattern Recognition*. Oxford University Press, 1995. 61, 72

V. Blanz et T. Vetter : A morphable model for the synthesis of 3d faces. *In International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH'99)*, 1999. 3, 94, 96, 108

C. Bregler, A. Hertzmann et H. Biermann : Recovering non-rigid 3d shape from image streams. *In Conference on Computer Vision and Pattern Recognition (CVPR'02)*, 2000. 44

A. Brun : *Manifolds in Image Science and Visualization*. Thèse de doctorat, Linköping University, 2007. 22

N. Brunet, F. Perez et F. De la Torre : Learning good features for active shape models. *In International Workshop on Subspace Methods (Workshop ICCV'09)*, 2009. 104, 105

J. Bruske, E. Abraham-Mumm, J. Pauli et G. Sommer : Head-pose estimation from facial images with subspace neural networks. *In International Conference on Neural Network and Brain (ICNNB'98)*, pages 528–531, 1998. 36

P. Buhlmann et B. Yu : Boosting with the l2 loss : Regression and classification. *Journal of the American Statistical Association*, 98:324–339, January 2003. 62

C.J.C Burges : *Data Mining and Knowledge Discovery Handbook : A Complete Guide for Researchers and Practitioners*, chapitre Geometric Methods for Feature Extraction and Dimensional Reduction : A Guided Tour, pages 1–34. Kluwer Academic Publishers, 2005. 22

R. Caruana : Multitask learning. *Machine Learning*, 28:41–75, 1997. 79

M. Chaumont et B. Beaumesnil : Robust and real-time 3d-face model extraction. *In International Conference on Image Processing (ICIP'05)*, pages 461–464, 2005. 44

Y. Chen et F. Davoine : Simultaneous tracking of rigid head motion and non-rigid facial animation by analyzing local features statistically. *In British Machine Vision Conference (BMVC'06)*, page II :609, 2006. 45

L.-F. Cheong et C.-H. Peh : Depth distortion under calibration uncertainty. *Computer Vision and Image Understanding*, 93:221–244, 2004. 48

S. Choi et D. Kim : Robust head tracking using 3d ellipsoidal head model in particle filter. *Pattern Recognition*, 41:2901–2915, 2008. 30

R.R. Coifman et S. Lafon : Diffusion map. *Applied and Computational Harmonic Analysis*, 21:5–30, 2006. 25

T. Cootes : *Image Processing and Analysis*, chapitre Model-Based Methods in Analysis of Biomedical Images, pages 223–248. Oxford University Press, 2000. 113

T. Cootes, G. Wheelera, K. Walkerb et C. Taylora : View-based active appearance models. *Image and Vision Computing*, 20:657–664, 2002. 133

T. F. Cootes, G. J. Edwards et C. J. Taylor : Active appearance models. *In European Conference on Computer Vision (ECCV'98)*, 1998. 108, 109

T. F. Cootes et C. J. Taylor : A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999. 93

T. F. Cootes et C. J. Taylor : Statistical models of appearance for computer vision. Rapport technique, Imaging Science and Biomedical Engineering, University of Manchester, 2004. 3, 96, 97, 99, 119

T. F. Cootes, C. J. Taylor, D. H. Cooper et J. Graham : Active shape models—their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. 45, 95, 96

T. F. Cootes, K. Walker et C. J. Taylor : View-based active appearance models. *In International Conference on Automatic Face and Gesture Recognition (FG'00)*, page 227, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0580-5. 94

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. 125

T.F. Cootes et C.J. Taylor : On representing edge structure for model matching. *In Conference on Computer Vision and Pattern Recognition (CVPR'01)*, page 1114, 2001. 49, 50, 97, 109

S. Cornou, M. Dhome et P. Sayd : Bundle adjustment : a fast method with weak initialisation. *In British Machine Vision Conference (BMVC'02)*, 2002. 43

T. Cox et M. Cox : *Multidimensional scaling.* Chapman and Hall, 2000. 23

D. Cristinacce et T. Cootes : Boosted regression active shape models. *In British Machine Vision Conference (BMVC'07)*, pages 880–889, 2007. 61

D. Cristinacce et T. Cootes : Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008. 3, 97

N. Dalal et B. Triggs : Histograms of oriented gradients for human detection. *In Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 886–893, June 2005. 21

S. Das : Filters, wrappers and a boosting-based hybrid for feature selection. *In International Conference on Machine Learning (ICML'01)*, pages 74–81, 2001. 60

J.G. Daugman : Complete discrete 2-d gabor transform by neural networks for image analysis and compression. *IEEE Transaction on Acoustics, Speech and Signal Processing*, 36:1169–1179, 1988. 20

D. De Ridder, O. Kouropteva, O. Okun, M. Pietikäinen et R. Duin : Supervised locally linear embedding. *In Joint International Conference on Artificial Neural Networks and Neural Information Processing (ICANN/ICONIP '03)*, 2003. 27, 28

V. de Silva et J. B. Tenenbaum : Global versus local methods in nonlinear dimensionality reduction. *In Neural Information Processing Systems (NIPS'02)*, 2002. 27

A. Del Bue, F. Smeraldi et L. Agapito : Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25:297–310, 2007. 44

D. F. Dementhon et L. S. Davis : Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995. 45

A.P. Dempster, N.M. Laird et D.B. Rubin : Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 39:1–38, 1977. 72

E. W. Dijkstra : A note on two problems in connection with graphs. *Numerische Mathematik*, 1:269–271, 1959. 24

R. Donner, M. Reiter, G. Langs, P. Peloschek et H ; Bischof : Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1690–1694, 2006. 108

F. Dornaika et J. Ahlberg : Fitting 3d face models for tracking and active appearance model training. *Image and Vision Computing*, 24(9):1010–1024, 2006. 109, 110, 120, 125

F. Dornaika et F. Davoine : Simultaneous facial action tracking and expression recognition in the presence of head motion. *International Journal of Computer Vision*, 76 (3):257–281, 2008. 45, 91, 107

H. Drucker : Improving regressors using boosting techniques. *In International Conference on Machine Learning (ICML'97)*, pages 107–115, 1997. 62, 63, 74

H. Drucker, C. Burges, L. Kaufman, A. Smola et V. Vapnik : Support vector regression machines. *In Advances in Neural Information Processing Systems (NIPS'86)*, 1996. 133

I. Dryden et K.V. Mardia : *Statistical Shape Analysis*. John Wiley & Sons, Inc., 1998. 90

S. Duffner et C. Garcia : A connexionist approach for robust and precise facial feature detection in complex scenes. *In International Symposium on Image and Signal Processing and Analysis (ISPA'05)*, 2005. 89

N. Duffy et D. Helmbold : Boosting methods for regression. *Machine Learning*, 47 (2-3):153–200, 2002. 63

G.J. Edwards, C.J. Taylor et T.F. Cootes : Interpreting face images using active appearance models. *In Conference on Automatic Face and Gesture Recognition (FG'98)*, pages 300–305, 1998. 55

M. Everingham et A. Zisserman : Identifying individuals in video by combining "generative" and discriminative head models. *In International Conference on Computer Vision (ICCV'05)*, pages 1103–1110, 2005. 3, 30, 90

M.E. Farmer, S. Bapna et A. K. Jain : Large scale feature selection using modified random mutation hill climbing. *In International Conference on Pattern Recognition (ICPR'04)*, 2004. 58

W. Finnoff, F. Hergerta et H. G. Zimmermann : Improving model selection by nonconvergent methods. *Neural Networks*, 6:771–783, 1993. 75

M. Fischler et R. Elschlager : The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22:67–92, 1992. 3, 91

R.A. Fisher : The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936. 28

F. Fleuret : Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004. 68

R.W. Floyd : Algorithm 97 : Shortest path. *Communications of the ACM*, 5(6):345, 1962. 24

R. Féraud, O. Bernier, J.-E. Viallet et M. Collobert : A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):42–53, 2001. 31, 33

W.T. Freeman et E.H. Adelson : The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991. 21

Y. Freund et R. Schapire : A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997. 21, 59, 61, 73

J. Friedman : Greedy function approximation : A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001. 62, 63

B. Fröba et C. Küblbeck : Robust face detection at video frame rate based on edge orientation features. *In International Conference on Automatic Face and Gesture Recognition (FG'02)*, page 342, 2002. 49

P. Gacon, Pierre-Yves Coulon et Gérard Bailly : Non-Linear Active Model for Mouth Inner and Outer Contours Detection. *In European Signal Processing Conference EUSIPCO'05*, 2005. 100

C. Garcia et M. Delakis : Convolutional face finder : A neural architecture for fast and robust face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1408–1423, 2004. 38

A. Gee et R. Cipolla : Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647, 1994. 5, 134

X. Geng, D.-C. Zhan et Zhou Z.-H. : Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B : Cybernetics*, 35(6):1098–1107, 2005. 27

A.S. Georghiades, P.N. Belhumeur et D.J. Kriegman : From few to many : Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001. 122

F. Girosi et T. Poggio : Networks and the best approximation property. *Biological Cybernetics*, 63:169–176, 1990. 71

L. Gond, P. Sayd, T. Chateau et M. Dhome : A 3d shape descriptor for human pose recovery. *In International Conference on Articulated Motion and Deformable Objects (AMDO'08)*, 2008. 21

S. Gong, S. McKenna et J. J. Collins : An investigation into face pose distributions. *In International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 265, 1996. 1, 23

J Gonzalez, F. De la Torre, R. Murthi, N. Guil Mata et E. Zapata : Bilinear active appearance models. *In Workshop on Non-rigid Registration and Tracking through Learning*, 2007. 98

J. Gonzalez-Mora, F. De la Torre, N. Guil et E. Zapata : Learning a generic 3d face model from 2d image databases using incremental structure from motion. *Image and Vision Computing (Accepted for publication)*, 2010. 3, 94

G. Goudelis, A. Tefas et I. Pitas : Automated facial pose extraction from video sequences based on mutual information. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(3):418–424, 2008. 29

N. Gourier, D. Hall et J. Crowley : Estimating face orientation from robust detection of salient facial structures. *In International Workshop on Visual Observation of Deictic Gestures (Pointing'04)*, 2004a. 2, 36, 40, 122

N. Gourier, D. Hall et J. L. Crowley : Estimating face orientation from robust detection of salient facial structures. *In FG Net Workshop on Visual Observation of Deictic Gestures (Pointing'04)*, 2004b. 57, 134

N. Gourier, J. Maisonnasse, D. Hall et J. Crowley : Head pose estimation on low resolution images. *In Multimodal Technologies for Perception of Humans*, LNCS, pages 270–280, 2007. 31, 33, 40, 80

R. Gross, I. Matthews et S. Baker : Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(1):1080–1093, 2005. 89, 98, 109

N. Grujic, S. Ilic, V. Lepetit et P. Fua : 3d facial pose estimation by image retrieval. *In International Conference on Automatic Face and Gesture Recognition (FG'08)*, 2008. 29, 43

G. Guo, Y. Fu, C.R. Dyer et T.S. Huang : Head pose estimation : Classification or regression ? *In International Conference on Pattern Recognition (ICPR'08)*, pages 1–4, 2008. 31, 35

I. Guyon et A. Elisseeff : An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. 59

C. G. Harris et M. J. Stepheds : A combined corner and edge detector. *In Alvey Vision Conference*, 1988. 21

R. Hartley et A. Zisserman : *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 49, 156

T. Hastie et R. Tibshirani : *Generalized Additive Models*. Chapman and Hall, 1990. 63

X. He, S. Yan, Y. Hu et H. J. Zhang : Learning a locality preserving subspace for visual recognition. *In International Conference on Computer Vision (ICCV'03)*, page 385, 2003. 26

N. Hu, W. Huang et S. Ranganath : Head pose estimation by non-linear embedding and mapping. *In International Conference on Image Processing (ICIP '05)*, 2005. 24, 36

C Huang, H. Ai et S. Lao : High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):671–686, 2007a. 31, 34

G. Huang, M. Ramesh, T. Berg et E. Learned-Miller : Labeled faces in the wild : A database for studying face recognition in unconstrained environments. Rapport technique, University of Massachusetts, Amherst, 2007b. 136

D.H. Hubel et T.N. Wiesel : Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154, 1962. 38

P. Huber : *Robust statistics*. Wiley, 2004. 62

T. Ishikawa, S. Baker, I. Matthews et T. Kanade : Passive driver gaze tracking with active appearance models. Rapport technique CMU-RI-TR-04-08, Robotics Institute, Carnegie Mellon University, 2004. 144

M. Jones et P. Viola : Fast multi-view face detection. Rapport technique TR-20003-96, Mitsubishi Electric Research Laboratory, 2003. 31, 33

M. J. Jones et J. M. Rehg : Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002. 41

K. Kähler, J. Haber et H.-P. Seidel : Geometry-based muscle modeling for facial animation. *In Graphics interface (GRIN'01)*, 2001. ISBN 0-9688808-0-0. 3, 92

D. Kalinkina : *Automatic human face tracking in video sequences*. Thèse de doctorat, INRIA, Université Pierre et Marie Curie, 2009. 106

D. Kalinkina, A. Gagalowicz et R. Roussel : 3d reconstruction of a human face from images using morphological adaptation. *In Computer Vision / Computer Graphics Collaboration Techniques and Applications (MIRAGE'07)*, 2007. 92

J. Kaminski, A. Knaan et M. Teicher : Head orientation and gaze detection from a single image. *In International Conference Of Computer Vision Theory And Applications (VISAPP'06)*, 2006. 134

K. Kähler : *3D Facial Animation- Recreating human heads with virtual skin, bones, and muscles*. VDM Verlag, 2007. 91

P. Kittipanya-ngam et T.F. Cootes : The effect of texture representations on aam performance. *In International Conference on Pattern Recognition*, 2006. 97

R. Kohavi et G.H. John : Wrappers for feature subset selection. *Artificial Intelligence*, 97 (1-2):273–324, 1997. 59

H.-S. Koo et K.-M. Lam : Recovering the 3d shape and poses of face images based on the similarity transform. *Pattern Recognition Letters*, 29(6):712–723, 2008. 44

M. A. Kramer : Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991. 93

V. Krüger et G. Sommer : Gabor wavelet networks for efficient head pose estimation. *Image and Vision Computing*, 20(9-10):665–672, 2002. 36

N. Kwak et C. H. Choi : Improved mutual information feature selector for neural networks insupervised learning. *In International Joint Conference on Neural Networks (IJCNN'99)*, 1999. 59

N. Kwak et C.-H. Choi : Input feature selection by mutual information based on parzen window. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:1667–1671, 2002a. 68

N. Kwak et C.-H. Choi : Input feature selection for classification problems. *IEEE Transactions on Neural Networks*, 13:143–159, 2002b. 68

N. Kwak, S.-I. Choi et C.-H. Choi : Feature extraction for regression problems and an example application for pose estimation of a face. *In International Conference on Image Analysis and Recognition (ICIAR'08)*, pages 435–444, 2008. 26, 28

M. La Cascia, S. Sclaroff et V. Athitsos : Fast, reliable head tracking under varying illumination : An approach based on registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):322–336, 2000. 3, 29, 30, 90

Y. Le Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel et D. Henderson : Handwritten digit recognition with a back-propagation network. *In Advances in neural information processing systems (NIPS'90)*, pages 396–404, 1990. 38

Y. Le Cun, L. Bottou, Y. Bengio et P. Haffner : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. 38

S. Le Gallou : *Détection robuste des éléments faciaux par Modèles Actifs d'Apparence.* Thèse de doctorat, Université de Rennes 1, 2007. 49, 97

C.-S. Lee et A. Elgammal : Body pose tracking from uncalibrated camera using supervised manifold learning. *In NIPS Workshop on Evaluation of Articulated Human Motion and Pose Estimation (EHuM'06)*, 2006. 27

H.-S. Lee et D. Kim : Tensor-based aam with continuous variation estimation : Application to variation-robust face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1102–1116, 2009. 98

J. A. Lee et M. Verleysen : Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomputing*, 67:29–53, 2005. 25

V. Lepetit et P. Fua : Monocular model-based 3d tracking of rigid objects : A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005. 49

L. Leyrit, T. Chateau, C. Tournayre et J.-T. Lapreste : Visual pedestrian recognition inweak classifier space using nonlinear parametric models. *In International Conference on Image Processing (ICIP'08)*, 2008. 60

C.-G. Li et J. Guo : Supervised isomap with explicit mapping. *In International Conference on Innovative Computing, Information and Control (ICICIC'06)*, pages 345–348, 2006. 26, 28

K.-C. Li : Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86:316–342, 1991. 26, 28

K.-C. Li : On principal hessian directions for data visualization and dimension reduction : Another application of stein's lemma. *Journal of the American Statistical Association*, 87:1025–1039, 1992. 26

S. Li, Z. Lei, Y. Zheng et Z.F. Wang : *Deformable Models*, chapitre Shape And Texture-Based Deformable Models For Facial Image Analysis, pages 91–131. Springer, 2007a. 108

S. Z. Li et Z. Zhang : Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1112–1123, 2004. 2, 31, 33, 65, 67

Y Li, S. Gong, S. Sherrah et H. Liddell : Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5):413–427, 2004. 31, 33, 35

Y. Li et W. Ito : Shape parameter optimization for adaboosted active shape model. *In International Conference on Computer Vision (ICCV'05)*, pages 251–258, 2005. 93, 95, 101

Z. Li, Y. Fu, J. Yuan, T. S. Huang et Y. Wu : Query driven localized linear discriminant models for head pose estimation. *In International Conference on Multimedia and Expo (ICME'07)*, pages 1810–1813, 2007b. 42

R. Lienhart et J. Maydt : An extended set of haar-like features for rapid object detection. *In International Conference on Image Processing (ICIP'02)*, 2002. 149

S.-W. Lin, T.-Y. Tseng, S.-Y. Chou et S.-C. Chen : A simulated-annealing-based approach for simultaneous parameter optimization and feature selection of back-propagation networks. *Expert Systems with Applications*, 34(2):1491–1499, 2008. 59

G. Little, S. Krishna, J. Black et S. Panchanathan : A methodology for evaluating robustness of face recognition algorithms with respect to changes in pose and illumination angle. *In International Conference on Acoustics, Speech and Signal Processing (ICASSP'05)*, 2005. 37

H. Liu et R. Setiono : A probabilistic approach to feature selection - a filter solution. *In Internat(ICML'96)*, pages 319–327, 1996. 58

H. Liu et L. Yu : Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge & Data Engineering*, 17:491–502, 2005. 2, 58, 59, 60

X. Liu : Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1941–1954, 2009. 93, 95, 98, 102

D. Lowe : Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. 21, 35, 49

B. Ma, W. Zhang, S. Shan, X. Chen et W. Gao : Robust head pose estimation using lgbp. *In International Conference on Pattern Recognition (ICPR '06)*, pages 512–515, 2006a. 20, 31

Y. Ma, Y. Konishi, K. Kinoshita, S. Lao et M. Kawade : Sparse bayesian regression for head pose estimation. *In International Conference on Pattern Recognition (ICPR'06)*, 2006b. 133

M. Malciu et F. Prêteux : A robust model-based approach for 3d head tracking in video sequences. *In International Conference on Automatic Face and Gesture Recognition (FG'00)*, page 169, 2000. 30

P. Martins et J. Batista : Accurate single view model-based head pose estimation. *In International Conference on Automatic Face and Gesture Recognition (FG'08)*, 2008. 5, 135

H. Mathieu et P. Fuchs : Les capteurs de localisation. *In* Philippe Fuchs, éditeur : *Le Traité de la Réalité Virtuelle : Interfaçage, immersion et interaction en environnement virtuel*, volume 2, chapitre les capteurs de localisation. Les Presses de l'École des Mines de Paris, 3 édition, 2006. 151

I. Matthews et S. Baker : Active appearance models revisited. *International Journal of Computer Vision*, 60(1):135–164, 2004. 109, 123, 125

R. Meiria et J. Zahavi : Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171:842–858, 2006. 59

H. Mercier : *Modélisation et suivi des déformations faciales : Applications à la description des expressions du visage dans le contexte de la langue des signes.* Thèse de doctorat, Université Paul Sabatier, 2007. 5, 109, 135

H Mercier, J. Peyras et P. Dalle : Toward an Efficient and Accurate AAM Fitting on Appearance Varying Faces. *In International Conference on Automatic Face and Gesture Recognition (FG'06)*, pages 363–368, 2006. 123

K. Mikolajczyk et C. Schmid : A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005. 21

M. Milgram, R. Belaroussi et L. Prevost : Multi-stage combination of geometric and colorimetric detectors for eyes localization. *In International Conference on Image Analysis and Processing (ICIAP'05)*, 2005. 129, 139

P. Mittrapiyanuruk, G. N. DeSouza et A. C. Kak : Calculating the 3d-pose of rigid-objects using active appearance models. *In International Conference on Robotics and Automation (ICRA'04)*, 2004. 94

A. Mokhber, C. Achard et M. Milgram : Recognition of human behavior by space-time silhouette characterization. *Pattern Recognition Letters*, 29:81–89, 2008. 21

J. Moody et C. Darken : Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1(2):281–294, 1989. 72

E. Murphy-Chutorian, A. Doshi et M.M. Trivedi : Head pose estimation for driver assistance systems : A robust algorithm and experimental evaluation. *In Intelligent Transportation Systems Conference (ITSC'07)*, pages 709–714, 2007. 21, 35

E. Murphy-Chutorian et M.M. Trivedi : Head pose estimation in computer vision : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:607–626, 2009. 9, 36, 80, 140

S. Nakariyakul et D. P. Casasent : An improvement on floating search algorithms for feature subset selection. *Pattern Recognition*, 42(9):1932 – 1940, 2009. 59

P. Negri, X. Clady, M. Milgram et R. Poulenard : An oriented-contour point based voting algorithm for vehicle type classification. *In International Conference on Pattern Recognition (ICPR'06)*, 2006. 49

M. H. Nguyen, J. Perez et F. de la Torre : Facial feature detection with optimal pixel reduction svms. *In International Conference on Automatic Face and Gesture Recognition (FG'08)*, 2008. 89

M.H. Nguyen et F. De la Torre : Metric learning for image alignment. *International Journal of Computer Vision*, 88(1):69–84, 2010. 3, 101, 105

Q. D. Nguyen et M. Milgram : Online active feature models for lip tracking. *In Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA'09)*, 2009. 89

A. Nikolaidis et I. Pitas : Facial feature extraction and pose determination. *Pattern Recognition*, 33:1783 – 1791, 2000. 134

D. Nister et H. Stewenius : Scalable recognition with a vocabulary tree. *In Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 2161–2168, 2006. 55

S. Niyogi et W. T. Freeman : Example-based head tracking. *In International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 374, 1996. 29, 30

M. Nordstrøm, M. Larsen, J. Sierakowski et M. B. Stegmann : The IMM face database - an annotated dataset of 240 face images. Rapport technique, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2004. 122

T. Ojala, M. Pietikainen et D. Harwood : A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29(1):51–59, 1996. 20, 52

T. Ojala, M. Pietikäinen et T. Mäenpää : Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. 52

J. Orozco, F.X. Roca et J. Gonzalez : Real-time gaze tracking with appearance-based models. *Machine Vision and Applications*, 20(6):353–364, 2009. 144

M. Osadchy, Y. Le Cun et M.L. Miller : Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, 2007. ISSN 1533-7928. 34, 38

I. S. Pandzic et R. Forchheimer, éditeurs. *MPEG-4 Facial Animation : The Standard, Implementation and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2003. 91, 92

C. Papageorgiou et T. Poggio : A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000. 21

J. Park et I. W. Sandberg : Universal approximation using radial-basis-function networks. *Neural Computation*, 3(2):246–257, 1991. 71

F. I. Parke : Computer generated animation of faces. *In ACM annual conference*, 1972. 3, 91

F. I. Parke : A parameteric model for human faces. Rapport technique, University of Utah, 1974. 91

F. I. Parke : Parameterized models for facial animation. *IEEE Computer Graphics and Applications*, 9(2):61–68, 1982. 91

K. Pearson : On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(6):559–572, 1901. 23, 28

H. Peng, F. Long et C. Ding : Feature selection based on mutual information : criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238, 2005. 59, 68

V. S. Petrovic et T. F. Cootes : Vehicle type recognition with match refinement. *In International Conference on Pattern Recognition (ICPR'04)*, pages 95–98, 2004. 49, 51, 52

M.J.D. Powell : Radial basis functions for multivariable interpolation : a review. *Algorithms for approximation*, pages 143–167, 1987. 73

L Prechelt : Early stopping – but when ? *In Neural Networks : Tricks of the Trade*, pages 55–69, 1998. 75

P. Pudil, J. Novovicova et J. Kittler : Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994. 59

L. Quan et B. Triggs : A unification of autocalibration methods. *In Asian Conference on Computer Vision (ACCV'00)*, pages 917–922, January 2000. 152

R. Rae et H.J. Ritter : Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks*, 9:257–265, 1998. 1, 35, 36

G. Rätsch, M. K. Warmuth, S. Mika, T. Onoda, S. Lemm et K.-R. Müller : Barrier boosting. *In Conference on Computational Learning Theory (COLT '00)*, pages 170–179, 2000. 63

B. Raytchev, I. Yoda et K. Sakaue : Head pose estimation by nonlinear manifold learning. *In International Conference on Pattern Recognition (ICPR'04)*, pages 462–466, 2004. 26

Y. Rodriguez et S. Marcel : Face authentication using adapted local binary pattern Histograms. *In European Conference on Computer Vision (ECCV'06)*, 2006. 52

C. A. Rogers : *Hausdorff measures*. Cambridge, University Press, 1999. 50

S. Romdhani, S. Gong et A. Psarrou : A multi-view nonlinear active shape model using kernel pca. *In British Machine Vision Conference (BMVC'99)*, 1999. 93

S. Romdhani et T. Vetter : Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. *In Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. 3, 99, 100

S. T. Roweis et L. K. Saul : Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000. 25, 28

H. A. Rowley, S. Baluja et T. Kanade : Rotation invariant neural network-based face detection. *In Conference on Computer Vision and Pattern Recognition (CVPR'98)*, page 38, 1998. 1, 31, 33, 34

A. Sattar, Y. Aidarous, S. Le Gallou et R. Séguier : Face alignment by 2.5d active appearance model optimized by simplex. *In International Conference on Computer Vision Systems (ICVS'07)*, 2007. 3, 94

A. Sattar, Y. Aidarous et R. Séguier : Gagm-aam : A genetic optimization with gaussian mixtures for active appearance models. *In ICIP*, 2008. 106

T. Sauquet, Y. Rodriguez et S. Marcel : Multiview face detection. Rapport technique IDIAP-RR 49, IDIAP, 2005. 31

B. Schiele et J. L. Crowley : Recognition without correspondence using multidimensiona lreceptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000. ISSN 0920-5691. 21

B. Schiele et A. Waibel : Gaze tracking based on face-color. *In Workshop on Automatic Face and Gesture Recognition*, 1995. 31, 33

B. Schölkopf, A. Smola et K.-R. Müller : Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998. ISSN 0899-7667. 24, 28

G. Schwarz : Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978. 61

S. Sclaroff et J. Isidoro : Active blobs : region-based, deformable appearance models. *Computer Vision and Image Understanding*, 89(2–3):197–225, 2003. 98, 108

T. Senechal, L. Prevost et S. Muhammad Hanif : Neural network cascade for facial feature localization. *In Artificial Neural Networks for Pattern Recognition*, 2010. 89

K. Sengupta, P. Lee et J. Ohya : Face posture estimation using eigen analysis on an ibr (image based rendered) database. *Pattern Recognition*, 35(1):103–117, 2002. 29

R. Séguier, S. Le Gallou, G. Breton et C. Garcia : Adapted active appearance models. *EURASIP Journal on Image and Video Processing*, 2009:1–14, 2009. 98

J Sherrah et S. Gong : Fusion of perceptual cues for robust tracking of head pose and position. *Pattern Recognition*, 34(8):1565–1572, 2001. 29

J. Sherrah, S. Gong et E.-J. Ong : Face distributions in similarity space under varying head pose. *Image and Vision Computing*, 19(12):807–819, 2001. 20

D. L. Shrestha et D. P. Solomatine : Experiments with adaboost.rt, an improved boosting scheme for regression. *Neural Computation*, 18(7):1678–1710, 2006. 61, 62, 63, 75

W. Siedlecki et J. Sklansky : A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10:335–347, 1989. 58

T. Sim, S. Bakern et M. Bsat : The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):1615 – 1618, 2003. 122

P. Simard, D. Steinkraus et J. Platt : Best practices for convolutional neural networks applied to visual document analysis. *In International Conference on Document Analysis and Recogntion (ICDAR'03)*, pages 958–962, 2003. 39

P. Somol, P. Pudil et J. Kittler : Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:900–912, 2004. 58

P. D. Sozou, T. F. Cootes, C. J. Taylor et E. C. Di Mauro : Non-linear point distribution modelling using a multi-layer perceptron. *In British Machine Vision Conference (BMVC'95)*, pages 107–116, 1995a. 93

P.D. Sozou, T.F. Cootes, C.J. Taylor et E.C. Di Mauro : Non-linear generalization of point distribution models using polynomial regression. *Image and Vision Computing*, 13:451–457, 1995b. 93

S. Srinivasan et K. L. Boyer : Head pose estimation using view based eigenspaces. *In International Conference on Pattern Recognition (ICPR '02)*, volume 4, 2002. 23

M.B. Stegmann et R. Larsen : Multi-band modelling of appearance. *Image and Vision Computing*, 21(1):61–67, 2003. ISSN 0262-8856. 97

R. Stiefelhagen : Estimating head pose with neural networks - results on the pointing04 icpr workshop evaluation data. *In Pointing'04 ICPR workshop*, 2004. 42

R. Stiefelhagen et J. Garofolo, éditeurs. *Multimodal Technologies for Perception of Humans, First International Evaluation Workshop on Classification of Events, Activities and Relationships, CLEAR 2006, Southampton, UK, April 6-7, 2006, Revised Selected Papers*, volume 4122 de *Lecture Notes in Computer Science*, 2007. Springer. 39, 80

P. Sturm : Algorithms for plane-based pose estimation. *In Conference on Computer Vision and Pattern Recognition (CVPR'00)*, pages 1010–1017, June 2000. 157

J. Sung, T. Kanade et D. Kim : Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, 80(2):260–274, 2008. 90

T. Svoboda, D. Martinec et T. Pajdla : A convenient multi-camera self-calibration for virtual environnements. volume 14, 2005. 152

M. J. Swain et D. H. Ballard : Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991. 21

X. Tan et B. Triggs : Enhanced local texture feature sets for face recognition under difficult lighting conditions. *In International Workshop on Analysis and Modeling of Faces and Gestures (AMFG'07)*, pages 168–182, 2007. 20, 51, 52

J. B. Tenenbaum, V. de Silva et J. C. Langford : A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000. 1, 24, 25, 28

D. Terzopoulos et K. Waters : Analysis of facial images using physical and anatomical models. *In nternational Conference on Computer Vision (ICCV'90)*, 1990. 91

D. Terzopoulos et K. Waters : Analysis and synthesis of facial image sequences using physical and anatomical models : 3-d modeling in image analysis and synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):569–579, 1993. 3, 91, 92

M. E. Tipping : Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001. ISSN 1532-4435. 108

C. Tomasi et T. Kanade : Shape and motion from image streams under orthography : A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 43

W. S. Torgerson : Multidimensional scaling. *Psychometrica*, 17:401–419, 1952. 23, 28

B. Triggs, P. F. McLauchlan, R. I. Hartley et A. W. Fitzgibbon : Bundle adjustment - a modern synthesis. *In International Workshop on Vision Algorithms (Workshop ICCV '99)*, 2000. 44

J. Tu, Y. Fu, Y. Hu et T. S. Huang : Evaluation of head pose estimation for studio data. *In Multimodal Technologies for Perception of Humans*, LNCS, pages 281–290, 2007. 2, 41, 42, 80

M. Turk et A. Pentland : Face recognition using eigenfaces. *In Conference on Computer Vision and Pattern Recognition*, 1991. 23

T. Tuytelaars et K. Mikolajczyk : Local invariant feature detectors : A survey. *Foundations and Trends in Computer Graphics and Vision*, 3:177–280, 2008. 21

M. Unser : Local linear transforms for texture measurements. *Signal Processing*, 11(1):61–79, 1986. 98

L. Vacchetti, V. Lepetit et P. Fua : Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004. 3, 90

H. Vafaie et K. DeJong : Robust feature selection algorithms. *In International Conference on Tools with Artificial Intelligence (ICTAI'93)*, 1993. 58

L. J. P. van der Maaten, E. O. Postma et H. J. van den Herik : Dimensionality reduction : A comparative review. Submitted to Journal of Machine Learning Research, 2009. 22

G. Van Dijck et M. Van Hulle : Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. *In International Conference on Artificial Neural Networks (ICANN'06)*, pages 31–40, 2006. 60

B. Van Ginneken, A.F. Frangi, J.J. Staal, B.M. ter Har Romeny et M.A. Viergever : Active shape model segmentation with optimal features. *IEEE Transaction On Medical Imaging*, 21(8):924–933, 2002. 95, 98, 102

M.A.O. Vasilescu et D. Terzopoulos : Multilinear independent components analysis. *In Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 547–553, 2005. 2, 41, 98

P. Viola et M. Jones : Rapid object detection using a boosted cascade of simple features. *In Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 1, 2001. 149

P. Viola et M. Jones : Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004. 21, 31, 57, 59, 65, 139

M. Vlachos, C. Domeniconi, D. Gunopulos, G. Kollios et N. KKollios : Non-linear dimensionality reduction techniques for classification and visualization. *In International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 645–651, 2002. 27

M. Voit, K. Nickel et R. Stiefelhagen : Neural network-based head pose estimation and multi-view fusion. *In Multimodal Technologies for Perception of Humans*, LNCS, pages 291–298, 2007. 2, 31, 33, 35, 39, 40, 80

P. D. Wasserman : *Advanced Methods in Neural Computing*. John Wiley & Sons, Inc., 1993. 73

K. Waters : A muscle model for animating three-dimensional facial expressions. *In International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH'87)*, 1987. 91

Y. Wei, L. Fradet et T. Tan : Head pose estimation using gabor eigenspace modeling. *In International Conference on Image Processing (ICIP'02)*, 2002. 20

K. Q. Weinberger et L. K. Saul : Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006. 25

K. Q. Weinberger, F. Sha et L. K. Saul : Learning a kernel matrix for nonlinear dimensionality reduction. *In International Conference on Machine Learning (ICML'04)*, page 106, 2004. 24, 28

A. Weissenfeld, O. Urfalioglu, K. Liu et J. Ostermann : Robust rigid head motion estimation based on differential evolution. *In International Conference on Multimedia and Expo (ICME'06)*, pages 225–228, 2006. 45

C. K. I. Williams : On a connection between kernel pca and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002. ISSN 0885-6125. 23

O. Williams, A. Blake et R. Cipolla : Sparse bayesian learning for efficient visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1292–1304, 2005. 108

M. Wimmer, F. Stulp, S. Pietzsch et B. Radig : Learning local objective functions for robust face model fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1357–1370, 2008. 4, 98, 103, 104, 109

L. Wiskott, J.-M. Fellous, N. Kuiger et C. Von der Malsburg : Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997. 57, 98, 133

B. Wu, H. Ai, C. Huang et S. Lao : Fast rotation invariant multi-view face detection based on real adaboost. *In International Conference on Automatic Face and Gesture Recognition (FG'04)*, pages 79–84, 2004. 31, 33

H. Wu, X. Liu et G. Doretto : Face alignment via boosted ranking model. *In Conference on Computer Vision and Pattern Recognition (CVPR'08)*, pages 1–8, 2008. 4, 65, 98, 105, 106, 109, 116, 125, 126

J. Wu et M.M. Trivedi : A two-stage head pose estimation framework and evaluation. *Pattern Recognition*, 41(3):1138–1158, 2008. 20, 24, 133

Y. Wu et K. Toyama : Wide-range, person- and illumination-insensitive head orientation estimation. *In International Conference on Automatic Face and Gesture Recognition (FG'00)*, page 183, 2000. 31, 33

J. Xiao, S. Baker, I. Matthews et T. Kanade : Real-time combined 2d+3d active appearance models. *In Conference on Computer Vision and Pattern Recognition (CVPR '04)*, volume 2, pages 535–542, June 2004a. 93, 94

J. Xiao, J. Chai et T. Kanade : A closed-form solution to non-rigid shape and motion recovery. *In European Conference on Computer Vision (ECCV'04)*, 2004b. 94

J. Xiao, T. Moriyama, T. Kanade et J. Cohn : Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85 – 94, 2003. 30

Z. Xiao, E. Dellandrea, W. Dou et L. Chen : ESFS : A new embedded feature selection method based on SFS. *In Advanced Concepts for Intelligent Vision Systems (ACIVS'09)*, 2009. 59

E. Xing, M. Jordan et R. Karp : Feature selection for high-dimensional genomic microarray data. *In International Conference on Machine Learning (ICML'01)*, 2001. 60, 61

J. Yan, S. Li, S. Zhu et H. Zhang : Ensemble svm regression based multi-view face detection system. Rapport technique MSR-TR-2001-09, Microsoft Research, 2001. 31, 33

S Yan, D. Xu, B. Zhang, H.-J. Zhang et S. Lin : Graph embedding and extensions : A general framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51, 2007. 26, 27, 28

J. Ybanez, F. Davoine et M. Charbit : A linear estimation method for 3d pose and facial animation tracking. *In Workshop on Component Analysis Methods for Classification, Clustering, Modeling and estimation Problems in Computer Vision (Workshop CVPR'07)*, 2007. 108

H. Yuan, S.-S. Tseng, W. Gangshan et Z. Fuyan : A two-phase feature selection method using both filter and wrapper. *In Conference on Systems, Man, and Cybernetics (CSMC'99)*, 1999. 58, 60

L. A. Zadeh : Soft computing and fuzzy logic. *IEEE Software*, 11(6):48–56, 1994. 69

R. Zemel et T. Pitassi : A gradient-based boosting algorithm for regression problems. *In Advances in Neural Information Processing Systems (NIPS'01)*, 2001. 62, 63

D. Zhang : *Image Retrieval Based on Shape*. Thèse de doctorat, Monash University, 2002. 21

L. Zhang, R. F. Chu, S. M. Xiang, S. C. Liao et S. Z. Li : Face detection based on multi-block lbp representation. *In International Conference on Biometrics (ICB'07)*, pages 11–18, 2007. 20

Z. Zhang : A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 152

Z. Zhang et H. Zha : Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, 2005. 25, 28

G. Zhao et M. Pietikäinen : Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:915–28, 2007. 52, 57

L. Zhao, G. Pingali et I. Carlbom : Real-time head orientation estimation using neural networks. *In International Conference on Image Processing (ICIP'02)*, pages 297–300, 2002. 33

S. K. Zhou, B. Georgescu, X. S. Zhou et D. Comaniciu : Image based regression using boosting method. *In International Conference on Computer Vision (ICCV'05)*, 2005. 61

Méthodes d'apprentissage pour l'estimation de la pose de la tête dans des images monoculaires

Cette thèse s'inscrit dans le cadre de PILE, un projet médical d'analyse du regard, des gestes, et des productions vocales d'enfants en bas âge. Dans ce contexte, nous avons conçu et développé des méthodes de détermination de l'orientation de la tête, pierre angulaire des systèmes d'estimation de la direction du regard.

D'un point de vue méthodologique, nous avons proposé BISAR (Boosted Input Selection Algorithm for Regression), une méthode de sélection de caractéristiques adaptée aux problèmes de régression. Elle consiste à sélectionner itérativement les entrées d'un réseau de neurones incrémental. Chaque entrée est associée à un descripteur sélectionné à l'aide d'un critère original qui mesure la dépendance fonctionnelle entre un descripteur et les valeurs à prédire. La complémentarité des descripteurs est assurée par un processus de boosting qui modifie, à chaque itération, la distribution des poids associés aux exemples d'apprentissage.

Cet algorithme a été validé expérimentalement au travers de deux méthodes d'estimation de la pose de la tête. La première approche apprend directement la relation entre l'apparence d'un visage et sa pose. La seconde aligne un modèle de visage dans une image, puis estime géométriquement l'orientation de ce modèle. Le processus d'alignement repose sur une fonction de coût qui évalue la qualité de l'alignement. Cette fonction est apprise par BISAR à partir d'exemples de modèles plus ou moins bien alignés. Les évaluations de ces méthodes ont donné des résultats équivalents ou supérieurs aux méthodes de l'état de l'art sur différentes bases présentant de fortes variations de pose, d'identité, d'illumination et de conditions de prise de vues.

Mots clés : pose de la tête, modèle déformable, alignement, sélection de descripteurs, régression, réseau de neurones incrémental, apprentissage automatique *boosting*

Learning-based head pose estimation in monocular images

This doctoral research is part of PILE, a medical project which aims at analyzing baby's gazes, gestures and vocalizations. In this context, we have designed and developed methods for determining the head pose which constitutes the cornerstone of a system for estimating the gaze direction.

From a methodological point of view, we have proposed BISAR (Boosted Input Selection Algorithm for Regression), a feature selection method which is well adapted to regression problems. It consists in iteratively selecting inputs of an incremental neural network. Each input corresponds to a feature selected by our Fuzzy Functional Criterion. The latter measures the functional relation between a feature and the values to predict. The features complementarity is provided by a boosting process that changes weight distribution on the training examples.

This algorithm has been experimentally validated in two head pose estimation methods. The first approach directly learns the relationship between the appearance of a face and its corresponding pose. The second approach aligns a face model in an image and then calculates the geometric orientation of this model. The alignment process is based on a cost function that evaluates the quality of the fitness. This function is learned by BISAR from examples of aligned and misaligned models. Evaluations of these methods have given state of the art results on different test sets with large variations in pose, identity, illumination and shooting conditions.

Keywords : head pose estimation, flexible model, alignment, feature selection, regression, incremental neural network, boosting, machine learning