Analysis, 3D reconstruction, & Animation of Faces

Charlotte Ghys

THÈSE

présentée pour l'obtention du titre de

DOCTEUR DE L'ÉCOLE NATIONALE DES PONTS ET CHAUSSÉES

Spécialité : Informatique

par

Charlotte GHYS

Analyse, Reconstruction 3D, & Animation du Visage

Analysis, 3D Reconstruction, & Animation of Faces

Soutenance le 19 mai 2010 devant le jury composé de :

Rapporteurs :	Maja PANTIC Dimitris SAMARAS
Examinateurs :	Michel BARLAUD Renaud KERIVEN
Direction de thèse :	Nikos PARAGIOS Bénédicte BASCLE

Abstract

Face analysis fields are widely spread out over a large quantity of domains : Human Computer Interaction, security, movies post-production, games... It includes detection, recognition, 3D reconstruction, animation, and emotion analysis. Face animation was the main motivation of this thesis, and we always kept it in mind at anytime.

We discuss here, most of the fields. We first talk about face reconstruction and face modeling with a new model inspired by the Candide Model. Then, we address face detection and particularly features detection introducing anthropometric constraints in a global scheme through a Markov Random Field formulation. From prior constraints, we are able to estimate the 3d pose of a face from a single image and to extend it to motion tracking. We conlude our work with emotion analysis. We propose an expression modelling technique defined as time series, and we present our 3D database for face expression. We present the state of the art of emotion recognition. And we finally invoke the use of our expression modelling technique as a prediction to be compared with the data in time.

Résumé

L'analyse du visage est un sujet très étudié dans de nombreux domaines : Interaction Homme Machine, sécurité, post-production cinématographique, jeux video... Cela comprend la détection, la reconnaissance, la reconstruction 3D, l'animation et l'analyse d'émotions. L'animation du visage a été la motivation principale durant toute la thèse. Nous nous intéressons à la plupart des domaines liés au visage : tout d'abord la reconstruction 3D et la modélisation de visage, avec un nouveau modèle de visage. Ensuite, nous introduisons des contraintes anthropométriques par champs de Markov pour la détection globale de points d'intérêts. Partant de contraintes anthropométriques, nous sommes capables d'estimer la pose 3D du visage à partir d'une seule image, et de l'étendre au suivi du visage. L'analyse d'émotion conclut notre travail : nous présentons une technique de modélisation d'expression définie comme une série temporelle et proposons de l'utiliser pour la prédiction d'émotions.

Acknowledgments

First of all, I would like to thank Nikos Paragios and Bénédicte Bascle who directed my researches during the different periods of my PhD Studies.

This thesis was accomplished with the support of Orange/France Telecom R&D (IRIS lab) and the CERTIS lab in Ecole Nationale des Ponts et Chaussées (ENPC), Paris. It has been funded by Orange and by the ANRT, the French national association for research and technology.

I thank the reviewers, Maja Pantic and Dimitris Samaras, for spending time reading my thesis and providing me helpful remarks.

I would like to express my thanks to Maja Pantic for welcoming me in her team at Delft University.

I would also like to thank Gérard Medioni for receiving me at USC, California in a student exchange program. In particular, I would like to thank Doug Fidaleo for his harmful welcome and a fruitful colaboration.

Several people have widely contributed to the work presented in this manuscript: Maxime Taron, Nikos Komodakis and Olivier Juan. I greatly thank all those collaborators and friends.

I would like also to thank some of my friends who supported and encouraged me during my thesis: at the CERTIS lab (Geoffray, Romain, Patrick), at the MAS lab (Salma, Lilla, Radhouène, Ahmed) and at the IRIS lab (Eric, Céline).

A special thanks goes to Olivier who always encouraged me during the difficult times. I will not have succeeded without his support. In this and many other things, I greatly thank him and acknowledge his patience.

Contents

List of Figures

Introduction

When we meet somebody for the first time and even before talking with him/her, the first thing that gives us information about a person, is the appearance in general, and particularly face. It can provide reliable information about the sex of the person, an approximation of his/her age, etc... Additionally, it could also inform us about the character or the emotional state of the person. A smiling person would be thought to be pleasant, while we think that somebody frowning is upset. For example the case of people with communication disabilities communicate not only with hands signs but also with facial expressions. The face is an essential part of their discussion, amplifying or downgrading the importance and the emotion of the ongoing conversation.
Faces, as well as expressions play really important part of communication, and are critical aspect in human computer interaction. Conventional means of communication between humans and machines/computers often rely on the use of a keyboard or a mouse. More advances techniques are based on emerging technologies from computer science and electronics, like for example the use of a touch screen. Such screens consist of more intuitive means of communication, in particular for people with limited familiarity with computers. One can find touch screen at ATMs, train stations when buying a ticket, or even at home, with graphic tablets. But this method, even if it improves the communication is convenient in one direction since it, doesn't permit the computer to answer the same way.
The most natural approach to interact with machines can be derived through the ways human communicate, that is based on facial expressions. The computer vision researchers for the past two decades have made significant progress on facial modeling and understanding. Current technologies allow to recover a 3D model of a the face, analyze it, generate an expression, etc ... It is natural to conclude that enabling to machines the ability to understand emotions of users could greatly influence human computer/machine interaction. Educational games are a good example ; the child is often accompanied by a virtual tutor (human, animal, or unreal character) to help him using the software. A system that is capable to capture and understand facial behavior could detect if the child is in difficulty. This could lead to an adjustment either of the speed or the difficulty of the game and the behavior

of the tutor.
Facial behavior is important tool in communication of deaf people. More and more researchers do focus on sign language and decoding it from videos. But they forgot the face movements, and their importance to help the understanding of the message and make it less ambiguous.
Human computer interaction is not the only field where analysis and reconstruction could be used. Image-based recognition is an emerging research and application area with enormous potential impact. Actually, Image Processing techniques can be used for identification and person recognition [121]. This aspect has been well studied through the use of frontal views/images. Face recognition performance through 2D images though heavily depends on the view point, the illumination conditions, etc. On the other hand, 3D facial recognition can be far more efficient since geometry - a fundamental component of our face - does not depend neither on the view point or the acquisition conditions. In the same field of research (safety/security/surveillance) analysis of body gestures is often considered to detect unusual behavior. It is natural to assume that this idea could be expanded to faces, and use analysis to detect when somebody's lying, or when the emotion which he/she expresses is fake or not.
Modeling faces, expressions include both spatio-temporal information and could serve in different tasks. To standardize the studies on faces, and particularly face behavior, norms were created. They define face key points, simple facial movements or associations of facial movements to describe expressions, etc. One can distinguish, among all standard, two, that are mostly used. Facial Action Coding System was created very early, during the 70's. It makes an inventory of all facial states which, when combined, can express any static emotion. Mpeg-4 is also a face standard, but with an animation purpose. It includes lists of face keypoints and of Facial Action Parameters, which combined, like facial states of Facial Action Coding System, can model any emotion in time. Those two standards will be detailed later. Once the importance of image-based facial modeling has been demonstrated, one has to deal with numerous technical and scientific challenges related with the task. This thesis aims on the use of image-based face modeling and understanding for human computer reconstruction. Such a process requires: (i) the creation of a generic 3D face model, (ii) the ability to infer the most probable variant of this models from a static or a sequence of images, and (iii) the inference of behavior using successive facial measurements (2D or 3D).

This thesis is organized as follows.
We first review reconstruction techniques, generic ones ([92, 17], ...), as face specific ones ([11, 97], ...). Then, we introduce our super resolution surface reconstruction approach [44] for face reconstruction. Such an approach is based on the use of emerging optimization tools like the use of discrete methods and

graph-cuts. Reconstruction result is so used as the target of a semantic face mesh for registration [46] and create an avatar of a given person. In order to eliminate the risk of correspondence between 3D models which is a rather tedious task, we consider a implicit representation approach which is parameter free and can deform freely surfaces while when available respecting facial landmarks constraints. Once the average model has been established, the next step consists of recovering a compact face representation which can also perform animations. To this end, we propose a Radial Basis Functions based Animation scheme to make the avatar alive.

Once the model has been determined, the next step consists of image-based inference. In the next thematic areas, we first review facial features extraction techniques. State of the art consists of appearance-based approaches ([104, 31], ...), geometric driven methods ([24, 75], ...) as well as hybrid solutions ([22, 112], ...), combining both appareance and geometry. Then, we introduce our work on facial features extraction and 3D pose estimation from a single image, this end, we combine prior geometric knowledge and machine learning techniques. First, we train several classifiers to detect the desired facial features in 2D images. This is done through the use of cascade adaboost. Once responses of the classifiers in the image provide a likelihood on the potential position of the facial landmarks, an MRF formulation is considered to perform inference. The response of the classifiers are used as singleton potentials, while the pair-wise interaction of this model, do encode anthropometric constraints and prior geometric knowledge on the relative positions of these points. The resulting framework is optimized using an efficient approach from linear programing, that is the primal-dual schema.We extend this to features tracking from a monocular sequence to retrieve their 3D positions in time. Once facial characteristics have been extracted from images, the next task consists of understanding and exploiting human emotions. Such a task can be performed based on facial features but the extraction of the related features is a rather tedious processing component. One of the most visible characteristics of emotion changes are facial changes. Therefore motion information can be a very valuable tool on understanding, modeling and inferring emotional states from images.

The next thematic area is dedicated to emotional analysis, understanding and inference from images. First, we review the state of the state of the art of emotion modelisation ([32, 39], ...). We propose a time-series approach that exploit facial feature motion for emotion characterization and emotion recognition. To this end, we introduce two variants of the model, one that aims to capture emotion smooth changes using a linear autoregressive model. Towards capturing more complex transitions, we also consider a non-linear approach using a neural network. Since these models heavily rely on the facial extraction step, we combine recognition with detection.we propose a connexionist approach [45] to predict features posi-

tion for a given emotion, and given initial positions. This prediction is applied with the purpose of emotion recognition. Where the other approaches extract information from data to classify them ([30, 101], ...), the prediction permits to make a comparison between expected data and real data leading to more accurate extraction and recognition as well.

Introduction (en français)

La première fois que l'on rencontre quelqu'un, avant même de lui parler, c'est son apparence et en particulier son visage qui nous donne des informations sur elle : son sexe, son âge approximatif, etc ... Son visage peut aussi nous renseigner sur son caractère, ou son humeur : une personne souriante sera considérée comme quelqu'un d'agréable ; alors qu'une personne qui fronce les sourcils serait plutôt en colère. Par exemple, les personnes mal-entendantes expriment beaucoup de choses à travers leur visage, en plus des signes qu'ils font avec les mains. Le visage occupe une part très importante dans leur discussions, en leur permettant de jouer sur l'intensité.

De manière générale, les visages aussi bien que les expressions jouent un rôle important dans toutes communications, et en particulier dans la communication homme-machine. Le plus souvent, les hommes interagissent avec les ordinateurs uniquement à l'aide d'une souris et d'un clavier. Des technologies plus avancées émergent comme les écrans tactiles : ils sont une manière plus intuitive de communiquer avec une machine, en particulier pour les personnes peu habituées aux ordinateurs. Ainsi on trouve de tels écrans un peu partout : dans les banques avec les distributeurs de billets, en gare pour acheter des tickets et même à la maison avec les tablettes graphiques. Même si cette technologie a fait ses preuves, elle ne permet pas à l'ordinateur de répondre de la même façon.

Une des façons les plus naturelles d'interagir avec les machines serait alors de copier une des manières dont les hommes communiquent entre eux : avec les expressions faciales. Durant les dix dernières années, les chercheurs en vision par ordinateur ont fait d'importants progrès sur la modélisation des visages et leur compréhension. Les technologies actuelles permettent d'obtenir le modèle 3D d'un visage, de l'analyser, de lui faire exprimer une émotion, etc ... Il paraît alors naturel de conclure que permettre à un ordinateur de décoder l'expression d'un visage ferait avancer d'un grand pas les interactions entre l'homme et la machine. Les jeux éducatifs en sont un bon exemple ; l'enfant est souvent guidé par un compagnon virtuel (humain, animal ou même imaginaire), pour l'aider à utiliser le logiciel. Un système qui serait capable de comprendre l'expression d'un visage, serait capable de détecter qu'un enfant est en difficulté. Cela permettrait

alors au logiciel d'adapter automatiquement la difficulté du jeu ainsi que le comportement du compagnon virtuel, par une nouvelle expression sur son visage et par des encouragements.
De la même manière, et nous l'avons déjà évoqué, l'expression faciale est un outil important dans la communication des personnes mal-entendantes. De plus en plus de chercheurs s'intéressent au langage des signes, en oubliant le plus souvent de tenir compte des mouvements du visage et de leur importance dans la compréhension de la conversation.
Mais l'interaction homme-machine n'est pas le seul domaine où le visage a son importance. Des méthodes en traitement d'image peuvent être utilisées pour l'identification et la reconnaissance de personne [121] dans le domaine de la sécurité et de la surveillance. Le plus souvent, cette reconnaissance se fait en utilisant des images de visages de face : le succès de cette reconnaissance repose pour beaucoup sur le point de vue, les conditions d'illuminations ... La reconnaissance de visage en utilisant la 3D pourrait être bien plus efficace puisque la géométrie du visage, qui définie ce même visage, ne dépend ni du point de vue ni des conditions d'acquisition. Dans ce même contexte, l'analyse des mouvements du corps est utilisée pour détecter les comportements inhabituels : cette idée pourrait être étendue simplement aux visages pour détecter par exemple si une personne ment ou si elle simule une émotion.
Pour harmoniser les études sur les visages, et en particulier sur les expressions faciales, des normes ont été définies. Elles spécifient les points d'intérêts du visage, les mouvements associés et les combinaisons de mouvements pour décrire des expressions. Parmi les différentes normes, plusieurs se détachent. Le Facial Action Coding System a été créé très tôt, durant les années 70. Ce système inventorie tous les états possibles du visage, qui, combinés les uns aux autres, permettraient de décrire n'importe quelle expression. Mpeg-4 est une autre norme, avec pour but de modéliser les mouvements du visage en vue de son animation. Mpeg-4 inclut les listes de points d'intérêt du visage, ainsi que les Facial Action Parameters, qui combinés, permettent de modéliser n'importe quelle émotion dans le temps. Ces deux normes seront détaillées plus tard. Maintenant que nous avons démontré l'importance de la modélisation de visage, il faut composer avec de nombreuses difficultés, liées à cette modélisation. C'est le but de cette thèse, et cela demande : (i) la création d'un modèle 3D de visage, (ii) la capacité de déformer ce modèle, à partir d'une image ou d'une séquence d'image pour modéliser un visage donné, et (iii) de déduire un comportement facial à travers différentes mesures, qu'elles soient 2D ou 3D.

Cette thèse est organisée de la manière suivante
Tout d'abord, nous présentons l'état de l'art des techniques de reconstruction, les méthodes génériques ([92, 17], ...) et les méthodes spécifiques aux visages

([11, 97], ...). Nous présentons ensuite notre méthode de reconstruction de surface en super résolution [44] appliquée à la reconstruction de visage. Elle est basée sur les derniers outils d'optimisation comme les méthodes discrètes et les graph-cut. La reconstruction ainsi obtenue est utilisée comme cible pour le recalage d'un modèle de visage [46] et permet d'obtenir l'avatar d'une personne. Pour éviter les erreurs de recalage entre le modèle 3D et une reconstruction, nous avons utilisé une approche basée sur une représentation implicite sans paramètres qui permet de déformer librement les surfaces, tout en respectant les contraintes liées aux points d'intérêt. Appliqué sur une base de données de reconstruction 3D de visage, le recalage permet ainsi d'obtenir un modèle moyen, qui sera la base pour obtenir ensuite l'avatar d'une personne donnée. L'étape suivante consiste enfin à définir un modèle d'animation compact tel que les fonctions radiales, qui nous permettent d'animer le visage de manière réaliste, avec un minimum de paramètres.
Ensuite, il convient de déterminer la position du visage ainsi que ses points d'intérêt à partir d'une image. L'état de l'art que nous présentons, peut être divisé en plusieurs catégories : les approches basées sur l'apparence ([104, 31], ...), les méthodes dites géométriques ([24, 75], ...) ou encore les solutions hybrides, combinant l'apparence et la géométrie du visage ([22, 112], ...). Nous introduisons alors notre travail sur l'extraction de points d'intérêt, et l'estimation de la pose 3D à partir d'une seule image, en combinant des connaissances à priori sur la géométrie et des techniques d'apprentissage. Nous avons entraîné plusieurs classifieurs en cascade (cascade Adaboost) pour détecter les points d'intérêt dans des images 2D. Le score des classifieurs fournit en chaque point de l'image la vraisemblance des points d'intérêt du visage. La position des points en est déduite via une formulation par champs de Markov. La réponse des classifieurs est utilisée comme une fonction potentielle ponctuelle alors que les interactions entre paires de points servent à coder les contraintes anthropomorphiques ou les connaissances à priori sur les positions relatives des points d'intérêts. Le champs de Markov est optimisé par une technique efficace de programmation linéaire suivant une approche primale-duale. Nous avons généralisé cette technique au suivi de points d'intérêt 3D dans des séquences vidéos monoculaires. Une fois les caractéristiques faciales extraites des images, nous pouvons les analyser en vue de les reproduire sur notre modèle ou de les reconnaître. De tels objectifs peuvent être atteints en se basant sur les points d'intérêt mais leur extraction est plutôt difficile : les expressions provoquant des mouvements parfois très discrets du visage. Ces informations de mouvement sont alors primordiales dans la compréhension, la modélisation ou la reconnaissance des émotions.
Enfin, la dernière étape de notre travail consiste en l'analyse, la compréhension et la reconnaissance des émotions à partir d'images. Dans un premier temps, nous rapportons l'état de l'art de la modélisation des émotions ([32, 39], ...). Ensuite nous proposons une approche basée sur les séries temporelles qui exploite le mou-

vement des points d'intérêt pour la modélisation et la reconnaissance d'émotions. Pour cela, nous introduisons deux variantes d'un même modèle auto-régressif permettant de capturer les changements subtils des émotions : le premier est un modèle linéaire alors que le deuxième est un modèle non linéaire défini par réseau de neurones afin de capturer des mouvements plus complexes et réalistes. Comme ces modèles reposent fortement sur l'extraction des points d'intérêt, nous avons combiné reconnaissance et détection [45] en prédisant la position des points d'intérêt pour une émotion donnée. Là où les autres approches extraient les informations des images afin de reconnaître les émotions par classification ([30, 101], ...), notre prédiction permet de faire une comparaison entre les valeurs attendues et observées pour en obtenir une extraction et une reconnaissance plus juste.

Chapter 1

Face Reconstruction and Face Modeling

In this chapter we focus on 3D facial reconstruction and statistical modeling. In the most generic case, multiple views can provide a rather precise geometric representation of the face through conventional stereo techniques. In the context of our research, we aim at understanding expressions, therefore the resolution of the model can be critical. Our first contributions focus on super-resolution reconstruction using multiple views through the use of discrete labeling. Once a number of subjects have been reconstructed, the next step consists of recovering a canonical face representation. Such a process requires customizing a generic model to a specific population. The most prominent way to address this task is through registration between the generic model and the set of available examples. Then, the registered examples can be used as basis to determine the most prominent model with respect to the given population. The second contribution of this chapters refers to an implicit variational framework for geometric registration using distance functions and thin plate splines. In order to improve accuracy as well as make the process more robust, we incorporate soft landmark correspondence constraints through the use of an additional energy component.

1.1 Introduction

In Computer Vision, 3D reconstruction is a very classical task, studied for several years now, and despite enormous advances, several issues still remain open. The aim of 3D reconstruction is to retrieve the 3D geometric representation of an object either from an image, several images, a sequence, etc ...It has been used in a number of application domains, with specific methods for buildings and town or body parts and in our case, faces. The goal of face reconstruction, is to

determine, as precisely as possible, the three dimensional geometry of the face, like for example a precise representation of the hollows of the eyes, the peak of the nose .... During the past two decades, the main focus of 3D face reconstruction was computer graphics, animation, and human computer interaction. More recently we have witnessed an important shift of this problem towards safety and security applications. Indeed, even if it is fairly easy to change the appearance of a person, it is much more challenging to change significantly the shape of a face, without surgery. Such an observation has motivated an number of academics as well as scientists working in industry to consider the 3D geometry of faces for recognition. Such a process requires recovering as precisely as possible the facial geometry of a given subject and the comparison with examples from a predefined data base. Let us first review existing techniques on stereo reconstruction.

There are many approaches for face reconstruction. Some of them require special hardware [53] like laser scanners, learning database [11], a model [43] or are only based on image(s).

1.2 Facial Reconstruction and State of the Art

1.2.1 Material Based Reconstruction

Laser range scanning

The most known and used method to recover geometric information is the Laser range scanning method. It consist in sending a laser ray which is reflected on the surface of the object to be reconstructed and returned to the system. The depth is then computed either by estimating the time between the ray departure and arrival, or by triangulation. The result is very accurate, probably the most accurate among all reconstruction methods, but the hardware is very expensive. For this reason, the method is usually considered towards database creation during a learning stage [11], and not directly for the reconstruction. Some examples of 3D scans of faces used in [11] are presented in [Fig. (1.1)]. To recvoer 3D geoIn order to overcome the limitation of cost, the idea of structural light projection has emerged during the last decade.

Structured light projection

To recover 3D geometry of an object, several laser rays are sent simultaneously on the object in the shape of stripes of different colors and are convolved with a sinusoidal function. The stripes deformations permit to compute the depth of each point on the surface. For example, the method proposed in [53] permits to reconstruct faces with structured light projection. Vertical color stripes (Red,

Figure 1.1: Some examples of the 3D Scan database used in [11].

green, blue) are successively projected to the subject, while the image is captured. The video (as well as the stripes projection) frame-rate is very high (up to 120 Hz), therefore we can assume that the subject doesn't move during the acquisition of the 3 frames (one for each stripes color). These three images can then be used to capture the 3D structure of the face. An example of such a hardware system is shown in [Fig.(1.2)]. Such a system in terms of performance provides a rather accurate reconstruction of the face, on the other hand, it requires the use of a rather complex hardware set-up of a relative important cost and therefore is not suitable for rather heavy use.

This is not the case for vision-based systems, simply because web-cams quite inexpensive of high-resolution are now available and offer a rather satisfactory frame rate. Last, but not least setting up such devices is straight forward even for unexperienced users. Coupling the use of such cameras with an intelligent software plugin that can perform facial reconstruction is an ongoing research effort, and could greatly shape a number of fields, like human-computer interaction, video communication and compression, biometrics, etc. A significant amount of research work has been devoted to this subject. In the upcoming section we will briefly review the state-of-the-art, discuss their limitations with respect to our objective and propose an alternative.

1.2.2 Generic reconstruction methods

The most common manner to retrieve the 3D geometry of objects and scenes using image is through image-based two-view reconstruction. The idea is rather simple : under the assumption that we can observe the projection of a 3D point in two different images taken from different view points, it's possible to determine a measurement that associates the observed images with the relative depth of this point. The same idea is the basis of multi-view reconstruction methods where additional views contribute to the reconstruction process through additional constraints. In order to review the two-view stereo-reconstruction methods, one first has to introduce some basic notions of image-based 3D geometry.

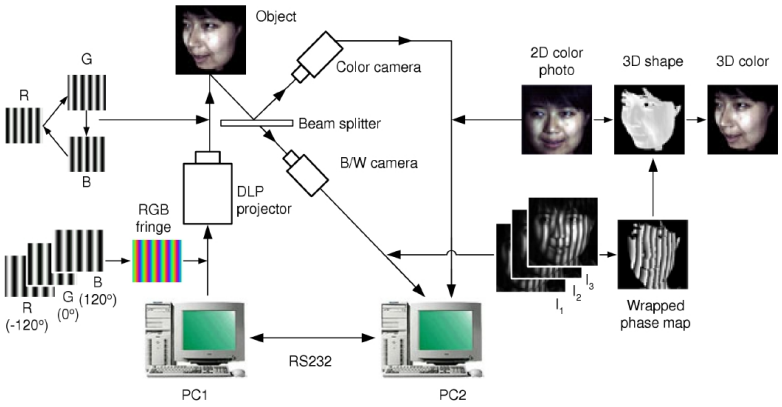


Figure 1.2: Structured light projection scheme proposed in [53].

Reminder on 3D Geometry

Let us consider an image taken from a specific camera positioned at a specific view point [Fig. (1.3)]. Such an acquisition involves a set of parameters: the intrinsic matrix A, encoding the internal parameters of the camera, such as

$$A = \begin{pmatrix} \alpha_u & \gamma & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}.$$

where α_u and α_v represent the focal length expressed in pixel units, (u_0, v_0) are the coordinates of the principal point c (where the optical axis pierces the image plane perpendicularly) and γ, the skew, usually equal to 0. Thus, the extrinsic matrix P that refers to the external parameters of the camera is defined as

$$P = A[Rt]$$

with R the rotation matrix and t the translation vector, give the position of the camera. Given a point M in 3D, its projection m on the image is given by :

$$m = PM$$

In the case of two cameras [Fig. (1.4)], the set-up can be described by the fundamental matrix F. F encodes the links between both cameras. Given a 3D point M, and its projection m and m' in the images I and I', the projection on I' of the ray L_m going through M and m is $l'_m = Fm$ and is called epipolar line. One should note that m' lies on this line. The intersection of all the epipolar lines

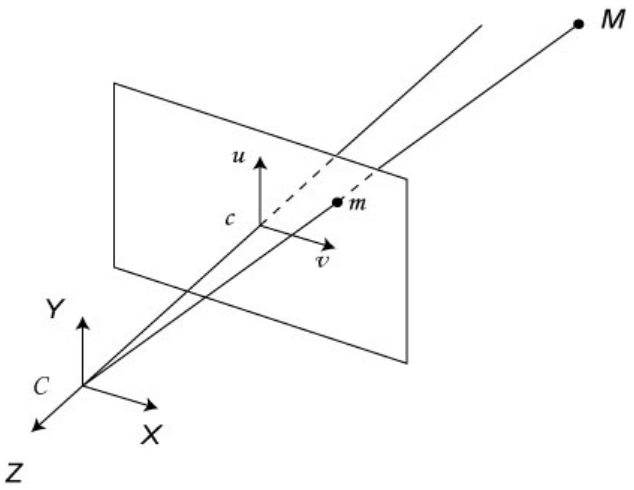


Figure 1.3: 3D Geometry in the case on one camera. The axes X, Y and Z, and the center C form the camera coordinates system while u, v and c form the image coordinates system. M is a given 3D point and m is its projection on the image according to $m = PM$ where P is the projection matrix, associated to the camera.

l is the epipolar center e. Knowing the coordinates of m and m' and the projection matrices P and P', it is possible to compute the coordinates of the 3D point M. Recovering the 3D shape of an object comes to the same thing, in stereo, than finding the pixel correspondences between the two images. Since the corresponding point m' of a point m is on the epipolar line and since the fundamental matrix F is known, this is a 1D problem. To simplify the matching the rectification process warps the images such that for $m = (x\ y\ 1)^T$, $m' = (x + d\ y\ 1)^T$, 3D geometry reconstruction remains a recovering d, where d is the disparity (See [Fig. (1.5)]). ovide certain individual measurements per image location regarding the depth of the corresponding 3D point. However, these measurements are individual and often quite unreliable in particular in the lack of image texture where establishing image-based correspondences become problematic. Therefore, work has been devoted on introducing global process to recover the complete geometry of the scene. In such a context, image-based stereo terms are combined with regularization constraints in terms of depth of neighborhood points towards addressing the ill-posed problem of the reconstruction. In the context of reconstruction, variational and level set methods, visual-hull based methods as well as MRFs and combinatorial optimization are the most popular ones.

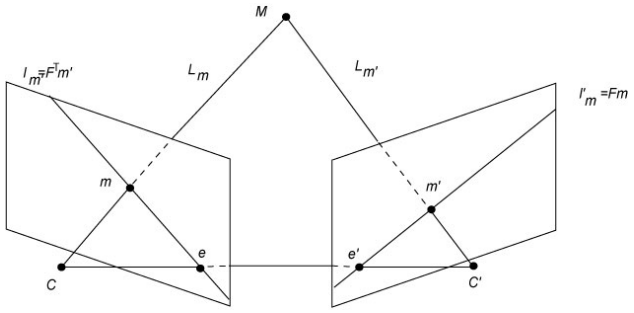


Figure 1.4: 3D Geometry in the case on two cameras of center C and C'. Given a 3D point M, its projections on images I and I' are respectively m and m' through rays L_m and $L_{m'}$. Projections of L_m and $L_{m'}$ are l_m and $l_{m'}$ and called epipolar lines. All epipolar lines intersect in the epipolar centers e and e'.

Variational and level sets methods

The central idea behind variational and level set methods is to deform a 3D surface under the influence of stereo-based images and regularization forces towards capturing the geometry of the scene. Such an approach can either be considered through a direct definition of the geometric flow deforming the surface or through the use of the gradient obtained through the minimization of a variational energy [62]. In both cases, the final step involves the deformation of an initial curve/surface in the normal direction (the tangential force does not change the geometry of the curve/surface but only its internal parameterization) under certain forces. The deformation of this surface can be implemented either explicitly or implicitly. Level sets methods [96] permit to make evolve a closed curve or surface $\underline{C}$ according to the equation

$$\frac{d\underline{C}}{dt} = \beta\vec{n}$$

driven by the movement β in the direction of its normal vector $\vec{n}$. In [92], the approach is based on surface deformation, such that the back projection of the surface on the images match. The authors reformulate the stereo problem as the minimization of a cost functional

$$\mathcal{E}_1 = \mathcal{M}_1(f) + \mathcal{R}_1(f)$$

where $\mathcal{M}_1(f)$ measures the statistical dissimilarity between the images and the back projected reconstruction, and $\mathcal{R}_1(f)$ defines regularizing constraints on f. Two similarity criteria were considered, separately, in two different forms. First criterion is simply the cross correlation. It measures the similarity between a

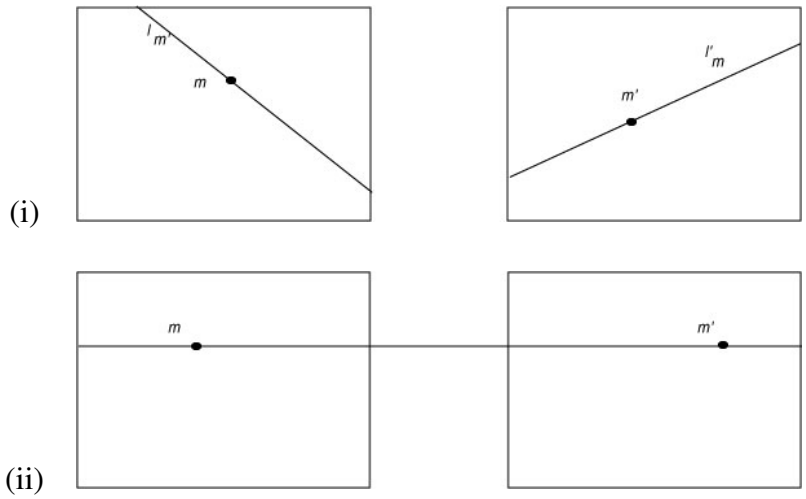


Figure 1.5: An example of images rectification : (i) The original images, (ii) the rectified images. m and m' are two corresponding points of coordinates $(x, y)^T$ and $(x', y')^T$ in the original images (i). The rectified images (ii) are wrapped such that m and m' coordiantes become $(x_r, y_r)^T$ and $(x_r + d, y_r)^T$ where d is the corresponding disparity.

pixel a in the first image and pixel a' in the second image. The second criterion is the mutual information which measures the statistical dependency of the images. Both criteria are computed globally in the entire image and locally on corresponding regions. The surface deformation is driven by the minimization of the energy functional through a gradient descent method. The same authors have extended their approach recently to encode different similarity metrics [92], as well as dynamic constraints. The method is extended in time, by estimating the 3D flow in a sequence [91]. Another related approach using variational level sets for the case of multi-view reconstruction was proposed in [117]. The idea there was to deform a 3D surface such that its projections in the corresponding image views separate the object from the background. Such an approach is more suitable in the absence of texture. The main limitation of variational and level set methods is their inability to capture the global minimum of the designed cost function. The use of gradient descend in high-dimensional spaces does often converge to a local minimum of the designed cost function. Discrete optimization is an alternative to gradient descent that can provide a better minimum.

Combinatorial methods

Unlike level sets methods, combinatorial methods [16, 64, 15] seek for the optimal solution not in the continuous space but a quantized version of it. Each solution is associated with a discrete label and the optimal label corresponds to the disparity that minimizes an energy similar to the one considered in the variational setting, or

$$\mathcal{E} = \mathcal{E}_{data} + \mathcal{E}_{smoothness}$$

where $\mathcal{E}_{data}$ is the data term, expressing the correlation between pixels in the images, and $\mathcal{E}_{smoothness}$ is the smoothness term constraining neighboring points to have approximately the same depth. The use of discrete methods has emerged due to the development of efficient optimization techniques in the context of computer vision. These methods rely on the max-flow/min-cut theorem [40]. One can refer to graph-cuts [17], efficient linear programming [65], tree-reweighted max-product methods [63] and belief propagation networks [116] are some examples.

Graph-cuts [17] are based on a graph $G = (V, E)$, where V is a set of nodes and E is a set of edges. One can imagine a node corresponding to a pixel while an edge encodes the link between pixels (neighboring pixels for example) using terms of the energy. The idea is to assign a label (in reconstruction case, it is the disparity) to each pixels. There are different ways to model the disparity computation with a graph. If we assume that disparities correspond to several labels, then we can either go with an incremental approach (α expansion) or one that aims to determine depth for all pixels at once. In the first case, we consider a subset of labels (two) and we solve the problem efficiently in that context. By randomly selected new labels against the current ones, and solving always the binary problem, one can claim that the final solution is a good minimum. The main advantage of this method is computational efficiency. The multi-way cut approach aims to determine the best label for every pixel at once through the max-flow principle. It has better guarantees in terms of the quality of the obtained minimum but is computationally expensive in particular in terms of memory and computer power requirements. Like in the case of variational techniques, these methods can be very efficient when dealing with texture images. On the other hand this is not the case when dealing with flat and smooth intensities. Convex-hull type approaches like space carving is an alternative in such a context.

Space Carving

Space carving consists in removing iteratively, from an initial volume V, parts until its back projection on the images corresponds to the images. This method was introduced in [66] for multi view reconstruction. Given an initial volume V

containing the scene and subdivided into voxels, the algorithm proceeds by iteratively removing voxels until it becomes identical to the maximal photo-consistent shape, V. (A shape is photo-consistent when the projection of its voxels on the camera corresponds to the image of the shape).

Structure from Motion

As previous methods are based on static image processing, structure from Motion refers to the process of building the 3D geometry of an object, moving in a video, while the camera is static, or of a static object and moving camera [100]. As in stereo, it is needed to compute the correspondence between points in the sequence. The major difference is that in stereo, images are grabbed at different points in space, while in structure from motion, images are captured at different points in time. The most efficient way to deal with the motion effects is through the separation between the rigid part of the scene and the non-rigid one while simultaneously recovering the internal camera parameters. Methods like factorization are the most frequently employed in this context.

Like most of the domains in computer vision, by taking into account specific application constraints one performs better inference between the images and the models. Facial reconstruction is an example where specific methods were introduced to cope with the problem. Based on the fact that facial geometry remains rather constant between individuals while at the same time one can point out certain nice properties like symmetry for example.

1.2.3 Specific Reconstruction methods

Face topology, common for all people, is not exploited in classical/generic reconstruction methods. Some of the reconstruction methods specific to faces, take advantages of face particularity using 3D face model, or asking user interaction. The other ones, exploit face databases and aim to built representations for the observed 3D face geometry.

Model-based Facial Reconstruction Methods

With the purpose of facial animation, Face Model Fitting consists in first computing the disparity map of the face and register a face mesh on it. In [43], stereo-driven approach to fit a model to the images is proposed. Starting with disparity maps from stereo images, the number of resulting 3D points is reduced and replaced by what they call "attractors", and their associated normals. As an option, automatically or manually selected feature points and silhouette are added to the previous data to enhance the results. A coarse face mesh is fitted to them with a

least squares method. The mesh is then refined several times, after a new adjustment to the data. The mesh is completed by a hair mesh, with a silhouette based fitting (due to the errors when computing the correspondent disparity map because of the possible lack of texture and contrast). This method is mostly a geometric one since it explores the depth maps. In [49], the same concept was considered but the method differs from [43] in terms of information that is used for inference. This method uses a video sequence, with a refinement of each frame. The system begins thus by the estimation of the head pose at each frame with a generic 3D face model, taking advantages of the symmetry of the face. The face model is globally deformed to adapt itself to the outer contours of the face and some of its internal features, minimizing the projection error in the video. It is finally locally deformed using a stochastic search optimization method. The process is integrated over an entire video sequence. These methods often require to know the camera parameters. In order to overcome this limitation, bundle Adjustment was introduced [18], a technique to refine 3D geometry of a scene and camera motions, simultaneously, from a given sequence of images.

Bundle Adjustment

In [42], bundle adjustment was considered to improve the model [40] fitting steps, using a model-driven bundle adjustment for an images sequences without requiring calibration data. To this end, the method uses face topology to extract the external camera parameters. To initialize the process, 5 points are manually selected in a frontal view of the face. As positions of those points in the generic face model are known, it is possible to deduce the position and the orientation of the camera.

In [97], a model-based bundle adjustment was considered, which, compared to model-driven bundle adjustment, uses the entire face model, not only a subset. The face model can be deformed in prefered directions, such that it is possible to make the nose longer, the face wider, etc ... by varying parameters defining the object. The authors propose to use two data terms for bundle adjustment : sematically meaningfull (or semantic) points on the object, and feature tracks, defined by a set of image points corresponding to a single object point. The problem becomes a minimization process, determining the parameters, by minimizing the projection of the semantic points and the feature tracks. While one can observe important facial deformations for certain part of the face, for a big part of the face the underlying motion is rigid. Therefore, structure from motion was also considered in this context.

Model-based Facial Reconstruction and Structure from Motion

In [37], classical structure from motion techniques for face reconstruction were amended to facial reconstruction, by introducing a deformable generic face model in the pose estimation stage and removing outliers using tensor voting [78]. They use a 3D face tracking, based on a 3D face model to derive the initial head pose estimate. This face model, roughly aligned gives first correspondences between identical points in optimal view selected frames. To refine these sparse correspondences, the best match is searched in a window around the first pixel. This permits to rectify images and obtain disparity map and then dense 3D point cloud. Finally, the outliers are removed, thanks to tensor voting technique. Most of the above methods assume a generic face model and seek for a projection of this model onto the image. However, human faces present variablity both in terms of geometry as well as in terms of texture. Therefore, when aiming generic face modeling it is natural to model these variabilities through statistical models.

Active Appearance Models

Active appearance models (AAM) [22] consist of building a face manifold from a training set of annotated texture images. It is a global approach and could be considered as Active Shape Model, where we added an appearance component. Once the training data are aligned by a procrustes analysis, the mean $\overline{\mathbf{x}}$ and $\overline{\mathbf{g}}$ and the matrices Q_s and Q_g depicting the modes of variation of, respectively, shape and texture, are computed. By varying the parameters c of the model, any face of shape x and texture g can be expressed using a linear combination of the modes of variation.

$$\mathbf{x} = \overline{\mathbf{x}} + Q_s c \tag{1.1}$$

$$\mathbf{g} = \overline{\mathbf{g}} + Q_g c \tag{1.2}$$

The main limitation of AAMs is the underlying Gaussian assumption on the distribution of shape and texture. This was addressed in [11] through a finer representation of both spaces.

Morphable Facial Models

Morphable models [11] are a very popular technique for face reconstruction. It refers to a generic framework which can achieve excellent results both in terms of synthesis as well as in terms of reconstruction. One can for example refer to the Mona Lisa's 3D reconstruction, from the famous Leonardo Da Vinci's painting. This paradigm is based on modeling textured 3D faces from only one or several

images and consists of a combination of a large number of 3D face scans, structured by a cylindrical representation, with radii $r(h, \phi)$ of surface points sampled at equally-spaced angles ϕ, and vertical steps h, and RGB values $R(h, \phi)$, $G(h, \phi)$ and $B(h, \phi)$. Using this base, and this representation one can create new instances of the model through a linear combination of the base. Computing the average face and the main modes of variation in the dataset, new shapes and new textures can be expressed in barycentric coordinates as linear combinations of the shapes and textures of the learning database faces. A probability distribution, is estimated for the coefficients of each mode from the example set of faces. This distribution enables to control the likelihood of the coefficients and is imposed on the morphing function to avoid unlikely faces. The reconstructed image has to be the closest to the input image in terms of Euclidean distances. The authors define shape and texture vectors that, added to or subtracted from a face, will manipulate a specific attribute while keeping all other attributes as constants as possible. This model is able to generate almost any face (applied to several person, the reconstruction reaches almost the quality of laser scans), as long as the face particularities are included in the database. Of course, the more elements are added in the database, the higher will be the number of the basis elements and the more challenging will be the estimation of the coefficients towards matching the data. This is mostly because the degrees of freedom of the model are keeping increasing while the inference constraints remain static. Introducing additional local geometric constraints is a natural direction to deal with this ill-posed process.

Facial Reconstruction, Priors and Image/Model-based approaches

In [69], the authors introduce a priori constraints for 3D stereo face reconstruction, based on the knowledge of low or high-curvature areas of the face. The reconstruction is based on the iteration deformation of a generic face model, by minimizing an energy function $E = \lambda_{ext} E_{ext} + \lambda_{int} E_{int}$. E_{ext} is a data term minimizing intensity difference between corresponding pixels in both images, while E_{int} is a regularization term homogenizing the depth of the vertices. Then, the mesh is totally transformed according to the a-priori knowledge on the curvature of some areas (minimum and maximum value the curvature of an area can take) and the position of known crest line such as the nose ridge.

Mixed reconstruction methods

In [73], the authors describe a technique, form a monocular sequence, which is also based on inter human variations. The face model used is represented by a linear combination of a neutral face and some number of vectors deforming linearly a face such as making the head wider or the nose bigger . . . The metrics coefficient

range and the interaction between points are manually determined. The authors proposed to determine face geometry using only two images from a sequence capture with a static camera and user interaction. 5 characteristic points on the face in those two images (inner corner eyes, mouth corners and nose tip) are manually selected to build a color model of the skin and compute the face position. One can note that with the actual progress in facial features extraction, (presented in the next chapter), those points can be automatically selected. Corners corresponding to high curvature points are computed and matched between both images. Once the correspondence are known, the head motion can be estimated easily and so, the 3D points positions are determined. The following step is the fitting of the face model, first, rough fitting using 3D reconstructed points, then, fine adjustment using image information. Actually, the whole reconstruction process can be seen as a mix between structure from motion, model fitting and inter-human variations based methods.

[120] and [68] introduced a face representation with four sets of parameters : shape, spherical harmonic basis, pose and illumination paramerters, whatever the illumination and the pose. The authors use this representation in [107] to recover the 3D shape of a face and its texture. From a set of silhouette images, the shape is recover following the visual hull technique. For a more accurate reconstruction, they introduce a shape model in the reconstruction step, constructed according to the Active Shape Model, from a learning 3D Face database. Then, using correspondances between the reconstruction and the input silhouette images and a spherical harmonic model constructed in the same way that the shape model, the authors recover the illumination and the spherical basis parameters. [112] presents a combination of 2D AAM [23] with 3D Morphable Models (3DMM) [11] to model faces. Actually, they prove that not only, 2D AAM can generate 3D phenomena, but also that they are able to model more invalid cases, than 3DMM. Thus, the authors explain how to constrain an Active Appearance Model, fitted on a sequence to compute the corresponding 3D shape modes (in a linear non-rigid structure from motion manner), so that it can only generate models can be represented by 3D faces. They finally extend their method for AAM fitting in real-time. The review of the state of the art leads to a natural conclusion for using prior models in the context of reconstruction. The advantage of such a concept is that models can be determined using either data-bases or conventional stereo-reconstruction techniques but once available one can retrieve 3D geometry using monocular sequences.

The next of this chapter is devoted to the model building process. First, we discuss how one can recover high resolution depth maps from low resolution sequences using combinatorial optimization. Once this task has been addressed, we present an implicit variational alignment method for 3D surfaces that can also ac-

count for the presence of landmark points. The last part of this chapter aims at determineinga low rank, compact 3D model that can be used both for animations and inference.

1.3 3D Face Reconstruction for Facial Animation

As mentioned in the state of the art, 3D face reconstruction can be achieved in many different ways. Our goal is to reconstruct face with the purpose to animate it. To this end, one should be able to have a parametric face model, (with animation parameters) representing the 3D geometry of the face. Model fitting [43, 49] appears to be the best way to reach this objective. To this end, we need to register a face model, to the 3D surface of the face. First, we will present our work on 3D face surface reconstruction, and then our animation face model and the registration technique to fit it on the surface.

Several approaches, which proved themselves, are available for surface reconstruction. For model fitting technique, there is no need of advance reconstruction, since it's the face model itself, which gives the human aspect of the reconstruction. Combinatorial methods are particularly suitable for this task, as the reconstruction process, in the case of rectified stereo images is very simple, and because the problem is directly defined as a discreet problem (compared to variational and level sets methods for which there is a need to reformulate the problem in the continuous domain). But this aspect is also a draw back when the reconstruction is based on low resolution images. We propose a Super Resolution Reconstruction [44], to improve the reconstruction.

1.3.1 Stereo Reconstruction and Graph Cut

Let us now introduce in details some notions from combinatorial optimization, namely the graph cut approach. Let G be a graph, consisting of a set of nodes $\mathcal{V}$ and a set of directed edges $\mathcal{E}$ that connect them such as $G = (\mathcal{V}, \mathcal{E})$ (See [Fig. (1.8).i]). The nodes set $\mathcal{V}$ contains two special terminal nodes which are called the source, s, and the sink, t, while the edges set $\mathcal{E}$ is divided in two sub-set : t-links for terminal links, linking terminal nodes with other nodes, and n-links for neighborhood links, linking non-terminal nodes together. All edges in the graph are assigned some non-negative weight or cost. A cut C is a partitioning of the nodes in the graph into two disjoint subsets S and T such that the source s is in S and the sink t is in T. The cost of a cut $C = (S, T)$ (See [Fig. (1.8).ii]) is defined as the sum of the costs of boundary edges (p, q) where $p \in S$ and $q \in T$. The minimum cut problem on a graph is to find a cut that has the lowest cost among all possible cuts. One of the fundamental results in combinatorial optimization is

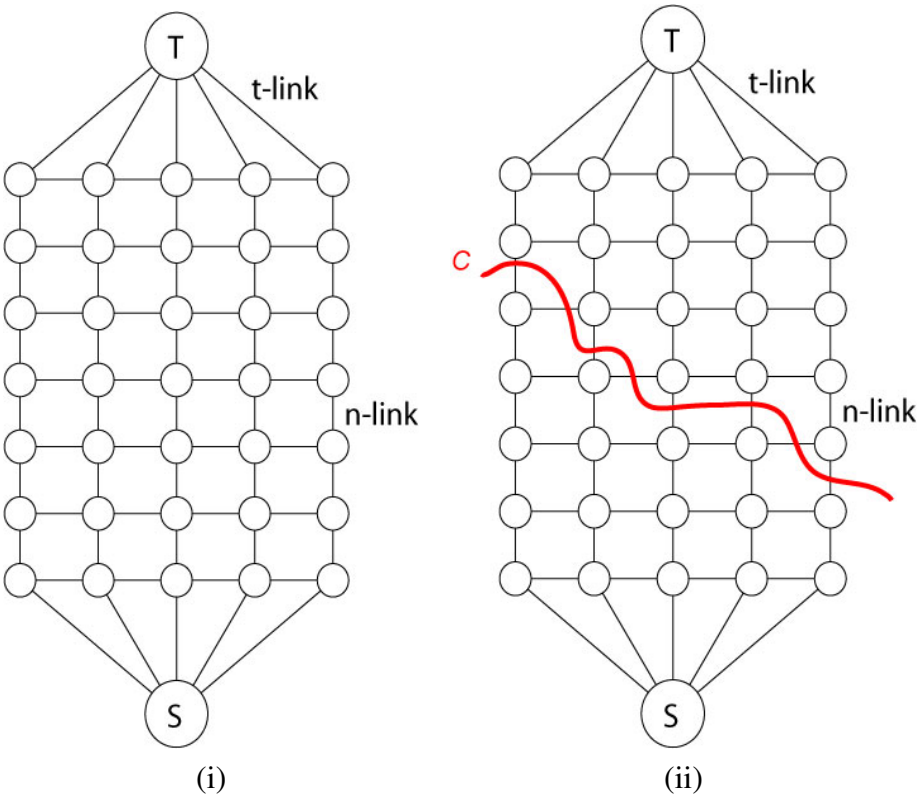


Figure 1.6: Example of a graph. (i) the graph $G = (\mathcal{V}, \mathcal{E})$, (ii) an example of cut. C a cut partitioning the set of nodes $\mathcal{E}$ in two subset S and T, containing respectively the source s and the sink t.

that the minimum cut problem can be solved by finding a maximum flow from the source s to the sink t.

Under certain conditions, one can prove that any optimization problem of the following form:

$$E = E_{data} + E_{smooth}$$

can be converted to a min cut/max flow problem. E represents the energy to minimize, and corresponds here the cost of the cut C. The definition of the data and smoothness terms depend on the problem to be solved. In the case of stereo reconstruction, the matching between intensities after applying the selected disparity component can be used as a data fidelity term. On the other hand smoothness often reflect the assumption that neighborhood pixels in the image correspond to the same depth level.

In a stereo-matching problem, the graph forms a 3D-mesh and a cut should be

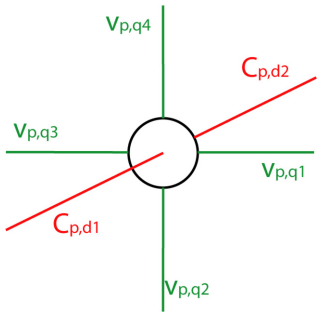


Figure 1.7: Six-connected vertex, in the case of stereo matching. In red, connections corresponding to the data term. In green, connections corresponding to the smoothness term.

view as a hyper surface. In this case, a vertex on the graph corresponds to a possible match between two images. Each of these vertices is six-connected (See [Fig. (1.7)]). Two connections, the ones in depth, correspond to other possible disparity values (connections of the same pixel in image 1 with neighbors along the epipolar line in image 2). The cost of these edges are $c_{p,d} = |I_1(x, y) - I_2(x + d, y)|$, the difference of intensity between pixel p with coordinates(x, y) in image 1, and pixel $(x + d, y)$ in image 2. It represents the data term of the energy we want to minimize. $E_{data} = \sum c_{p,d}$ where one end of $c_{p,d}$ is in S and the other one is in T. The four other edges are connections with the four neighbors in the image introducing a smoothness connectivity. As a scene is most of time considered as piecewise smooth, their cost is usually defined for simplicity by a Potts model $v_{p,q} = u_{p,d} \cdot T(d_p \neq d_q)$ where d_p and d_q are disparities of pixels p and q, and

$$u_{p,d} = \begin{cases} 2K \text{ if } |I_p - I_q| \leqslant U \\ K \text{ if } |I_p - I_q| > U \end{cases}$$

Such a potential function tolerates certain discrepansies between depth levels within local neighborhood while penalizing heavily more important deviations. And so $E_{smooth} = \sum v_{p,q}$ where q is in S and p is in T. However this $v_{p,q}$ definition ca nnot be used for face reconstruction, as a face can not be considered as a piecewise constant scene. Obviously because it is one and only one object with a continuous surface and not a scene, but also because it is made of curves and there are just few crests. Furthermore a difference of pixel intensity doesn't mean there is a difference of depth (an example of this is the skin space between eyebrows. It is very light compared to eyebrows, but the depth is the same). A good alternative to E_{smooth} term is given in the following section.

One should note that, as mentioned earlier, stereo matching with graph-cut can also be formulated and solved with α-expansion. Anyway, when minimizing the same energy, the results are the same.

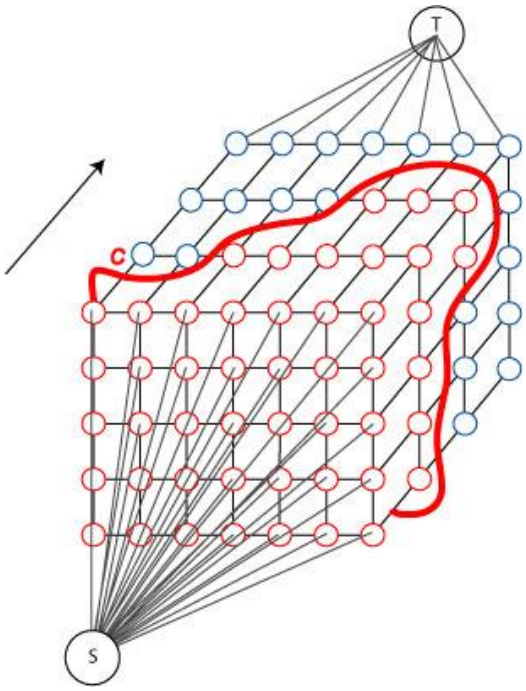


Figure 1.8: Example of a 3D graph for a stereo-matching problem. The arrow represents the direction of the increasing disparity, and C represent the cut partitioning the wet of edges $\mathcal{E}$ in two sub-sets S and T.

Redefinition of Local Consistency toward Exploiting Facial Geometry

As mentioned earlier the Potts model, used to define E_{smooth}, is not well designed for face reconstruction and must be redefined. To renforce E_{data} term, $v_{p,q}$ is derived from the matching cost in depth of the two vertices that it links.

$$v_{p,q} = k(c_{p,d} + c_{p,d+1} + c_{q,d} + c_{q,d+1})$$

It depends on the cost to assign the same disparity for both pixels. A larger k increases the smoothness of the reconstruction and avoids the outliers.

Such a global approach, with the redefinition of the smothness term, can lead to consistent 3D face models. But its performance depends on the resolution of input images. Model accuracy is important in a number of applications where input images suffer from low resolution. The use of multiple stereo pairs taken from the same camera set-up in time can be used to provide more accurate reconstructions. The idea behind such an approach is that due to the discretization process each stereo pair will be able to capture different parts of the 3D surface. Putting such temporal reconstructions together under the assumption of correspondences between images can lead to more precise depth maps.

Super Resolution Image Reconstruction

In computer vision applications, like medical, surveillance or satellite imaging, high resolution images are often required. Such images correspond to important pixel density where images are more detailed. On can find in [84] a good overview of different super-resolution techniques. The basic idea of all methods is to use multiple low resolution (LR) images captured from the same scene. These LR images are different representations of the scene, and considering there is no significant change between them, they are only shifted from each other with a sub pixel precision. Furthermore, in the grabbing process of images, there is a loss of information, due to the distortion of the camera, noise or blur. So, one can assume that capturing consists of transforming a high-resolution to a low resolution image and can be written as follows :

$$y_k = DB_kM_kx + n_k$$

where y_k is the k^{th} low resolution image of the sequence, x the high resolution image, D the decimating matrix, B_k the blur matrix, M_k the warping matrix representing the motion that occurs during image acquisition and n_k the noise vector. Most of the SR image reconstruction processes consist of three steps : registration, interpolation and restoration. Registration refers to the estimation of motion. Since the shifts between LR images are not regular, the registered low resolution images, will not always correspond to a uniformly spaced high resolution grid. Thus, non-uniform interpolation is necessary to obtain a regular high resolution image. Finally, image restoration is applied to the up sampled image to remove blur and noise. In this case, super resolution approach is only use to retrieve the sub-pixelic structure of the face and not to reconstruct it in term of image intensity. So those two last processes on the image, removing blur and noise, are not necessary.

Super Resolution Method

Usually, in disparity computation process, it is assumed that the disparity range is discretized to one pixel. To improve the sharpness of the results, the idea presented here is to use a disparity range discretized to a half pixel. This means, working with a disparity interval $[d_{min}, d_{max}]$ of size D, we would like to refine the disparity map assuming a disparity interval $[d'_{min}, d'_{max}]$ of size $F_m \times D$. Considering this, this is like multiplying the image width by a magnification factor of $F_m = 2$, one can consider that "'new pixels"' appear. Intensity values have then to be assigned to them. A first and obvious idea would be to interpolate the intensities of the neighboring pixels. But it supposes that the texture varies homogeneously. To avoid this false assumption, super resolution image reconstruction technique is used to computed the intensity of the new pixels.

In a general way, Optical Flow [51] refers to the motion field in an image. Given $I(x, y, t)$ the intensity of pixel (x, y) of image I at time t, using Taylor series, we have :

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

Assuming the intensity of a point doesn't change in time :

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

and,

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

Given $\frac{dx}{dt} = u$ and $\frac{dy}{dt} = v$, the optical flow constraint equation is :

$$-\frac{\partial I}{\partial t} = \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

where u and v are the motion respectively in x and y direction.

Optical flow is used here to estimate the sub pixel image structure by computing the direction and speed of object motion from one image to another, while we assume that the face movements are small between subsequent images.

Here, the first frame of the sequence is used as a reference frame. Considering we want to increase the disparity discretization, and thus increasing the number of correspondences along the epipolar lines, it comes to the same thing as apllying super resolution process in only one dimension in the image (See [Fig. (1.9)]) and to add nodes in the reconstruction graph in depth, between each existing nodes. Finally, the intensity of every new pixel is computed using a weighted nearest neighbor approach [2] of the shifted pixels in the subsequent images (for which the positions are computed by optical flow). The cost $c_{p,d}$ is so redefined:

$$c_{p,d} = I_1(x, y) - I_2(x + d, y)$$

with

$$\begin{cases} I(x,y) &= I_{LR,0}(\frac{x}{F_m}, y) & \textbf{if } x \textbf{ is even} \\ I(x,y) &= \sum_{k=0}^{p} \pi_k J_k & \textbf{else.} \end{cases}$$

where $I_{LR,t}$ is the LR image of the sequence at time t, $J = \{J_0, J_1, \ldots, J_p\}$ is the set of the p nearest neighbors of pixel(x, y) among shifted image $I_{LR,t}$, and π_k is the weight inversely proportional to the distance to the pixel(x, y).

[Fig. (1.10)] and [Fig. (1.11)] show the results of usual disparity maps (i) and of the corresponding super resolution disparity maps (ii). Four stereo couples

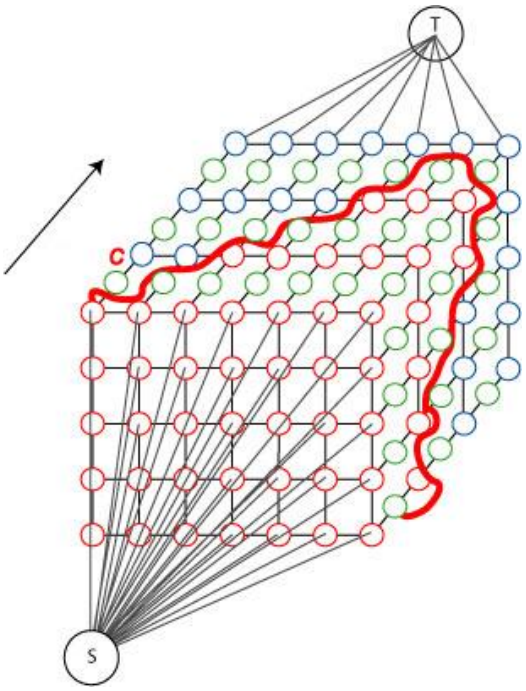


Figure 1.9: Example of a 3D graph for a stereo-matching problem using Super Resolution. The green nodes correspond to the added nodes between each existing nodes, and decreasing the disparity discretization to a half pixel.

images are used here with a magnification factor of 2. One can see that the super resolution reconstruction shows off the eye-socket and the crest of the nose. More images could be used for better results as well as a bigger magnification factor.

To test our method, and for each image of our dataset, we divided height and width by 2, and we perform super-resolution reconstruction on the resulting image. One should keep in mind that the super-resolution was not applied on the image to recover the original size, but on the graph, to discretized the disparity range to a half pixel. Thus comparing the super-resolution disparity map with the original image disparity map doesn't make any sense. Only a visual inspection allows to see the details of the construction appearing in the super-resolution reconstruction.

1.4 Face Model

The construction of a generic 3D face model, which can capture the geometric variation across individuals is a rather critical aspect of a method aiming to

reproduce facial animations. Such a model should be generic enough, involve a small number of parameters (control points) and be capable of reproducing realistic facial animations. Furthermore, one should be able to easily determine the correspondences/projections of these control points in the image plane.

1.4.1 Candide Face Model

Candide model is a parametrized face model for face analysis and facial animation. The original Candide [Fig. (1.12.i)] was created in 1987, by Rydfalk [94] and defined by 75 vertices and 100 triangles. In this version, only face was taken into account, while in the following one [Fig. (1.12.ii)], the neck was included in the model. The third and last version [Fig. (1.12.iii)], was actually an adaptation of the first one, where the number of vertices (and so triangles) is increased, around important face features, as the mouth, the eyes, and the nose.

The first reason why we decided to improve the original Candide model, is its global aspect, which makes think of an artificial face more than a human being, with its large and flat polygons. Then, we had no clue on the origin of the Candide model. It seems to us that the Candide model was totally manually created and not based on real faces. Such a simple model is not sufficiently realistic. This lead us to introduce real examples in the process to update the model and account variations across individuals [46].

1.4.2 Candide Improvements

In order to determine a realistic/generic 3D face model, we first modify the Candide model to increase the definition in regions like cheeks or forehead. Then, we considered the 3D RMA range dataset benchmark [1]. This acquisition system was based on structured light. The data we use are the raw reconstruction. No post-processing was done. The database consists in the face reconstruction of 120 people with different head orientations. Approximately 80 people where students with the same ethnic origins and nearly the same age. The rest consists of academic people, aged between 20 and 60. Among the 120 people, only 14 are women. For time reason and data quality (some acquisitions were made with person wearing glasses or facial hair), we only use 30 reconstructions. The database contains measurement errors and a high data variance. The image is defined as a volume whose pixels' value is equal to the distance of these pixels to the nearest laser point. Once such a data set has been determined, the next step consists of registering all training examples to the same reference pose. The Candide model has been considered as 'reference' pose configuration. Then, each example has been registered to this Model using the landmarks-enforced approach of the method proposed in [55] with thin plate splines [12] being the transformation domain.

In such a context both the source and the target shapes $C_{\mathcal{S}}, C_{\mathcal{T}}$ are represented using a distance transform [13] in a bounded domain Ω, or

$$\phi_{\mathcal{C}}(\mathbf{x}) = \begin{cases} 0, \ \mathbf{x} \in C_{\mathcal{C}} \\ d(\mathbf{x}, \mathcal{S}), \ \mathbf{x} \notin C_{\mathcal{C}} \end{cases}, \ \phi_{\mathcal{T}}(\mathbf{x}) = \begin{cases} 0, \ \mathbf{x} \in C_{\mathcal{T}} \\ d(\mathbf{x}, \mathcal{T}), \ \mathbf{x} \notin C_{\mathcal{T}} \end{cases} \tag{1.3}$$

with $d(\mathbf{x}, \mathcal{S})$ being the minimum distance between $\mathbf{x}$ and $C_{\mathcal{S}}$. In [55], the idea of measuring the local deformations between two shapes through a direct comparison of the quadratic norm of their distance functions (once the transformation has been determined) was proposed:

$$E_{dt}(\mathcal{L}) = \iiint_{\Omega} \chi_{\alpha}(\phi_{\mathcal{C}}(\mathbf{x}))(\phi_{\mathcal{C}}(\mathbf{x}) - \phi_{\mathcal{T}}(\mathbf{x} + \mathcal{L}(\mathbf{x})))^2 d\mathbf{x} \tag{1.4}$$

with χ_{α} being the characteristic/indicator function:

$$\chi_{\alpha}(d) = \begin{cases} 1/(2\alpha) & \text{if } d \in [-\alpha, \alpha] \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

The objective function (1.4) can be used to address global registration as well as local deformations. We use Thin Plate Spline (TPS) transformation [12] to address both. Let us consider N control points $\mathbf{\Theta} = \{P_i\}_{i=1}^{N}$ located on the surface of the source shape $\mathcal{S}$. Thin plate spline is then defined as the $\mathcal{C}^2$ transformation $\mathcal{L}$ minimizing the bending energy under constraints on the displacement of control points:

$$E_{sm}(\mathcal{L}) = \iiint_{\Omega} \|H_{\mathcal{L}}(\mathbf{x})\|_2^2 d\mathbf{x}, \ \text{ subject to } \mathcal{L}(P_i) = P_i', \tag{1.6}$$

where $\|H_{\mathcal{L}}(\mathbf{x})\|_2$ is the Froebenius norm of the Hessian matrix of the transformation and P_i' is the new position of the control point P_i.

The minimum of this functional verifies the biharmonic equation and may be written with the form

$$\mathbf{x}' = \mathcal{L}(\mathbf{x}, A, T, V_i) = A \cdot \mathbf{x} + T + \sum_{i=1}^{n} V_i U(\|P_i - \mathbf{x}\|), \tag{1.7}$$

where $A \cdot \mathbf{x} + T$ represents the affine part of the transformation and the set of vectors $\{V_i\}_{i=1}^{n}$ the weight of the non affine warping. $U(r)$ is a radial basis function, solution of the biharmonic equation (in 3D $U(r) = -|r|$). Given the considered representation of the model (19 critical feature points), it is natural to enforce correspondences between them across individuals. Therefore, assuming an annotated

database (done either manually or automatically), we introduce an additional data term, which aims to preserve correspondences between the annotated points:

$$E_{ld}(\mathcal{L}|\mathbf{m}^{\mathcal{C}}, \mathbf{m}^{\mathcal{T}}) = \sum_{i=1}^{19} \left|\left|\mathcal{L}(\mathbf{m}_i^{\mathcal{C}}) - \mathbf{m}_i^{\mathcal{T}}\right|\right|^2 \tag{1.8}$$

with $\mathbf{m}_i^{\mathcal{C}}$, $\mathbf{m}_i^{\mathcal{T}}$ being the i^{th} marker on the candide model and on the target face. One can now combine the image, smoothness and landmarks-based terms towards an objective function able to produce dense local and global correspondences, or

$$E(\mathcal{L}|\mathbf{m}^{\mathcal{C}}, \mathbf{m}^{\mathcal{T}}) = E_{dt}(\mathcal{L}) + \beta E_{sm}(\mathcal{L}) + \gamma E_{ld}(\mathcal{L}|\mathbf{m}^{\mathcal{C}}, \mathbf{m}^{\mathcal{T}}) \tag{1.9}$$

The lowest potential of this cost function can be determined using a gradient descent method. The last issue to be addressed is the rigidity of the transformation. TPS transformations are affine invariant, which, in the context of facial registration, is not natural. One can overcome this limitation using a gradient descent on the manifold of rigid transformations.

Examples of registration between the Candide model and the training set are shown in [Fig. (1.13)], while the final model is obtained through averaging of the 120 locally registered examples and is shown in [Fig. (1.14)]. One should point out that the training examples refer to faces at a neutral state.

Once the model is defined, it is so possible to reconstruct the 3D shape of any face, and register this new face model on it [Fig. (1.15)]. Then, the last step for a compleete face model is the definition of the animation parameters.

As the registration include a part of manual intervention, we could notice that the Candide Model was a lot more away from a human face target than our new face model. No visual inspection was done, as there were no expert involved in face analysis or recontruction in any of the 3 labs where the thesis was done. One should note that our principal goal was to present a new method to obtain a human being like face model, and not to define the perfect face model, as such a model would have asked a lot more time (and the objective of the thesis was to address most of the steps from image capture to face animation, and not only a face model).

1.4.3 Face Model Animation

State of the art

To obtain a realistic animation, only moving the feature points is not enough. It is necessary to compute the position of the other vertices of the mesh. Facial animation begins in 1972, with the key-framing introduced in [85]. A library of

facial expression was built, according to a face model (here a mesh). Each expression is represented by a key-frame at its most intensive moment. The animation is produced by interpolation between a face in a neutral state and one of the key-frame of the database. This approach, although very simple, is the starting point of facial animation.

In [105], the authors simulate muscles with a hierarchical spline modeling system based on local area surface deformation they call Langwidere. The multi-level shape representation provides more control over the possible deformations and speed the process while the bicubing B-splines offer more smoothness ans flexibility. Animation of a facial model is specified by setting the activation level of muscle groups. The effect of muscle activation is simulated by reproducing the local surface deformation caused by muscle contraction. The muscle is defined by an insertion point (the end of the muscle fixed to the skin), an attachment point (the end of the muscle fixed to the bone) and an area of influence, the portion of surface affected by the contraction (This system was introduced before in [108]). The method takes into account 3 muscle types (among a wide range of muscle types) the biomechanical characteristics (like flesh and muscles slide on a layer of fat over the bones and cartilage that give structure to the face) and the change of position of insertion and attachment points when muscles contract.

The system described in [59] simulates also muscle action, but using Free Form Deformations (FFD). FFD is a technique for deforming solid geometric models in a free form manner. Simply, FFD corresponds to deformations applied to control points of an imaginary 3D grid where the object to deform is embedded. The muscle action is simulated as the displacement of the control points of the control-unit for a FFD defined on a region of interest. To simulate the muscle action on the skin surface of human face, regions are defined on the face mesh which correspond to the anatomical description of the facial region on which a muscle action is desired. A parallelepiped control unit can be defined on the region of interest. The deformations which are obtained by acting muscles are simulated by displacing the control point and by changing the weights of the control points of the control-unit. The region inside the control-unit deforms a flexible volume, corresponding to the displacements and the weights of the control points. In order to propagate the deformations of regions to the adjoining regions, linear interpolation can be used to decide the deformation of the boundary points. The physical properties of the skin surface such as mass, stiffness and elastic characteristics can also be incorporated when applying the deformations.

The author decomposed face movements in minimal perceptible action (MPA) as basic motion parameter. Each of them is constituted of one or several simulated muscle actions. Each MPA has a corresponding set of visible facial features such as movement of eyebrows, movements of jaw, or mouth and others which occur

as a result of contracting and pulling of muscles associated with the region. Every expression is made up of one or more MPA's.

In [79], an approach is presented to create deformations of polygonal models using Radial Basis Functions (RBFs) to produce localized real-time deformations. Radial Basis Functions assume surface smoothness as a minimal constraint and animations produce smooth displacements of affected vertices in a model. The ability to directly manipulate a facial surface with a small number of point motions facilitates an intuitive method for creating facial expressions for virtual environment applications.
Face mesh geometry is locally deformed by a geometric deformation element (GDE). A GDE consists of a control point (does not have to be a vertex of the mesh), the region of influence around the control point, anchor points that lie on the boundary of the influence region (and surround the control point) and an underlying RBF system. The movable control points and the stationary anchor points determine the displacement of the vertices in the influence region such as :

$$y(x) = \sum_{i=0}^{N} w_i \rho(||x - c_i||) \tag{1.10}$$

where x is the initial position of the vertex, $y(x)$ is its new position, $\rho()$ is the radial basis function and c_i is the i^{th} element of the set containing the anchor points and the control point, while w_i are the associated weight. The region of influence is bound by a distance metric that determines the stationary anchor points. Two different distance metrics are proposed in the paper to specified the anchor point. The first one is based on edge depths in the tree: the control (or the closest mesh vertex) becomes the root for a search tree of mesh edges. Then we search down the tree of mesh edges with a Breadth First Search, determining all vertices within a specified depth. Leaf nodes of the search tree become the anchor points. The other one is simply based on Euclidean distance. In many cases, both metrics produce similar deformations. To animate the face, the new position of the control point has to be specify, and the RBF system computes the new locations of all vertices in the influence region based on the new control point position and the stationary anchor points.

1.4.4 Animation process of our model

The animation of a face model consists of estimating a number of parameters that explain the current state of the face model. Such an action requires a range of parameters explaining different movements [46]. The selection of the model is critical in this process in terms of performance, quality and computational cost.

One can conclude that a compromise between complexity and performance has to be made, aiming at a model that is able to reproduce the most critical deformations.

MPEG-4 specifies 84 Features Points (FPs) on the neutral face [Fig. (1.16)]. The main purpose of these FPs is to provide spatial references to define Facial Action Parameters (FAPs), corresponding to a particular facial action. Some FPs such as the ones along the hairline are not affected by FAPs. However, they are required to define the shape of a proprietary face model using FPs. FPs are arranged in groups such as cheeks, eyes and mouth. The location of these FPs has to be known for any MPEG-4-compliant face model.

The model proposed by MPEG-4 is still quite complex, while the estimation of the actual positions of the control points through image inference is quite problematic.

We selected 19 critical features among the 84 MPEG-4 FPs. The point was actually not to define exactly 19 points but to use the points that are sufficient to define or reproduce the 6 basic emotions as we'll define them later. In the context of low-pass band information transfert, the useless information should not be taken into account. We could have used points in the middle of the cheeks or the chin. But considering face hair on one hand, and lack of information in these region, it would have been difficult to detect them in a reasonnable time.

The selection of these points is guided from the potential representation of expressions using geometric deformations, as well as hardware acquisition constraints. The final model [Fig. (1.17.i)], consists of 19 degrees of freedom and refers to a simple 3D polygonal approximation of the face. Such a selection produces a reasonable compromise between complexity and performance. The position of each of these 19 points is easy to define on the Candide model [Fig. (1.17.ii)]. This is the starting point of our model.

The way our model is defined, anything but the position of the 19 feature points are known at each time. A muscle's simulation requires it to translate the feature points movement in terms of muscle's contraction or laxity. A more direct method would be to define directly the position of the other vertices of the mesh according to the feature points movement.

Building an animation model based on muscles motion is really a tough task. For realist movements, face and skull have to be known and reproduced perfectly. Actually, only a face mesh, modeling the skull, skin, tissu and muscles could reach this goal. Since our model is a mesh, this approach is not feasible. A method for mesh deformation like the one presented in [79] is more suitable to the model. To summarize, the deformation can be expressed such as :

$$y(x) = \sum_{i=0}^{N} w_i \rho(||x - c_i||)$$

where x is the initial position of a vertex, $y(x)$ is its new position, the c_i are the set defined by the anchor points and the control point, w_i are the associated weights, estimated in a learning phase, and $\rho()$ is a radial basis function. The way the influence area is defined is critical. The authors define the influence area according to a metric. They experienced two metrics. The first one considered the mesh as a tree where the control point is the root [Fig. (1.18.i)]. The vertices whose the depth is below a pre-defined threshold are defined as under the influence of the control point. Leaf nodes of the search tree become the anchor points. This method depends totally on the density of the mesh. For a dense mesh, the threshold have to be high, to obtain an influence area large enough. On the contrary, the threshold of a sparse mesh have to be small. The second one the author considered is the euclidean distance [Fig. (1.18.ii)]. This supposes that the shape of the influence area for all control points is a circle. But in reality, a control point is not necessary in the center of its influence area, particularly for the control points at the lips center. Such an influence area would not make the mouth opening possible.

To avoid the drawbacks of those two metrics, and considering that the face model is always the same (same vertices and same triangle), the influence areas are defined manually, once for all. Additionally, in the proposed method, all the control points lie on the mesh. This implies, in the case of mouth opening, one of the anchor point is at the lower border of the mesh, and as a consequence, it is impossible for the chin to go down. So we add external points surrounding the mesh, defined as the duplication of border vertices, moved away from the border [Fig. (1.18.iii)]. Thus the points on the mesh border are not static anymore.

To determine the new position of vertices, a weighted sum of RBF is associated to the control points, based on the control point and vertices describing the influence area, as in [Fig. (1.18)]. The weights c_i are estimated in a learning phase, when the face is in a neutral state.

[Fig. (1.19)] shows the mouth deformation with (i) and without (ii) texture, in details.

1.5 Conlusion

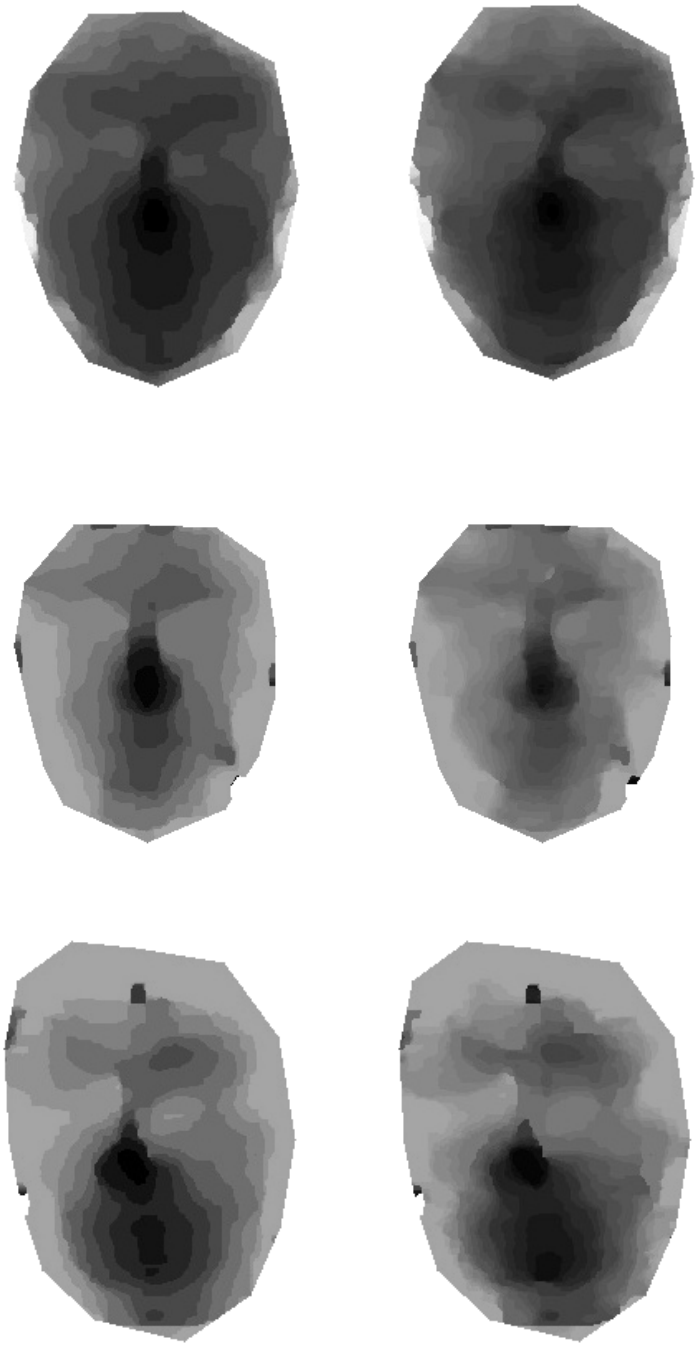
We presented in this first chapter the 3D reconstruction of a given face, from a stereo pair of images, with the purpose of animation.

First, a generic face model [46] was created based on the well known Candide model. To improve it, the mesh was refined and registred to a database of 30 people. The resultant and final 3D face model is the mean of all this registration. One can consider that a model based on 30 people is not enough and furthermore, that the difference between male facial topology and female facial topology should lead to two different face models. But the definition of a face model is not com-

plete without animation parameters. As the simulation of muscles, flesh and bones is not feasible in a reasonnable time for a given person, we chose to study the mesh deformation following feature points motion [46]. The mesh points influenced by these motions was manually selected, and learning step from a database should be envisaged.

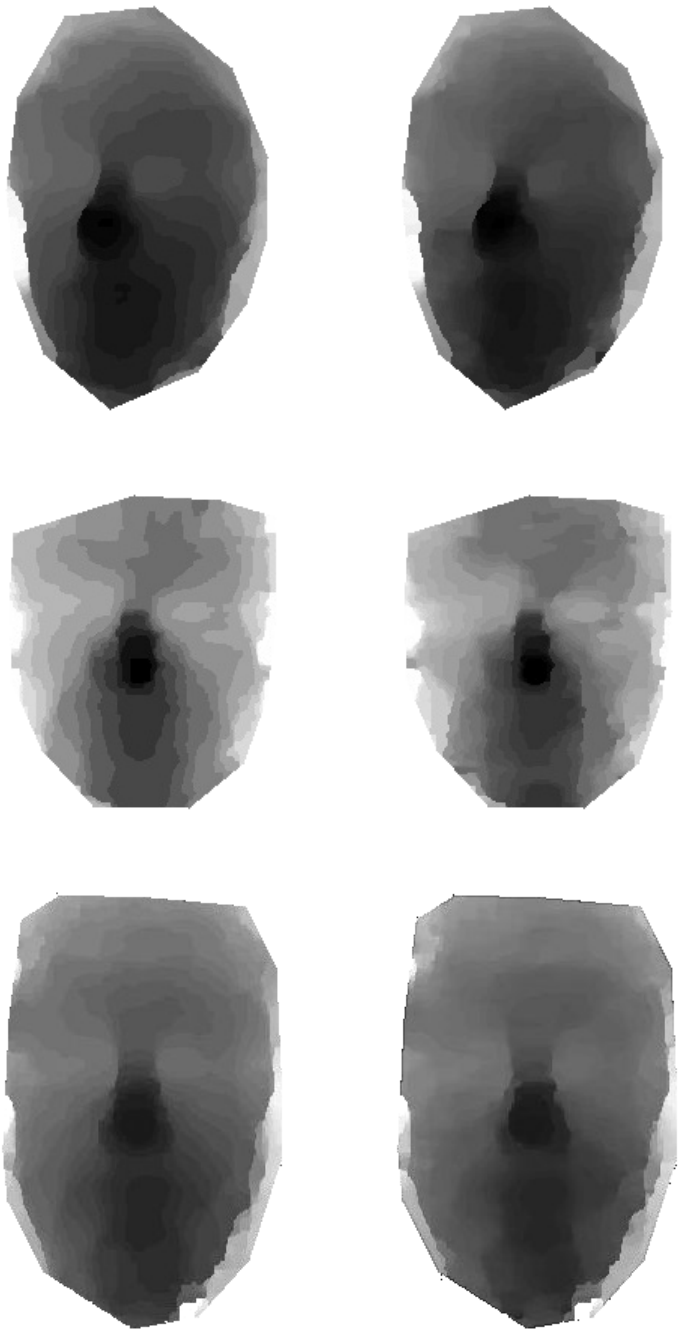
In parallel, from a stereo pair of images, we established a super resolution disparity map [44], with a disparity discretization decrease to a half pixel, to deal with low resolution images. The generic face model is then registered to the face surface reconstruction using Thin Plate Spline.

The next aspect to be addressed is the automatic detection and tracking of control points : for face reconstruction registration to our model on one hand, and for facial movements mimicking from a person to our model on the other hand.



(i) (ii)

Figure 1.10: Some examples of (i) disparity maps and (ii) the corresponding super resolution disparity maps.



(i) (ii)

Figure 1.11: Other examples of (i) disparity maps and (ii) the corresponding super resolution disparity maps.

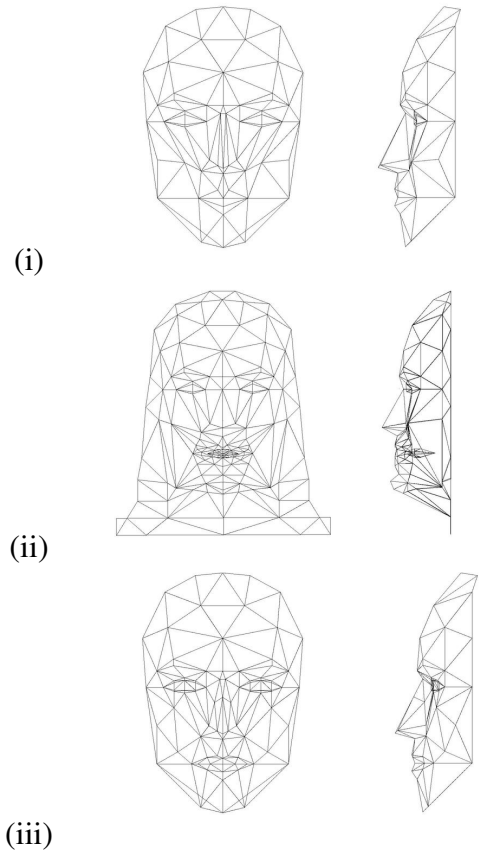


Figure 1.12: Evolution of the Candide Model. (i) The original Candide created in 1987 by Rydfalk. (ii) The second version of the Candide model by Bill Welsh in 1991 at British Telecom. (iii) The third version of the Candide model, developped by Jörgen Ahlberg in 2001 to simplify animation.

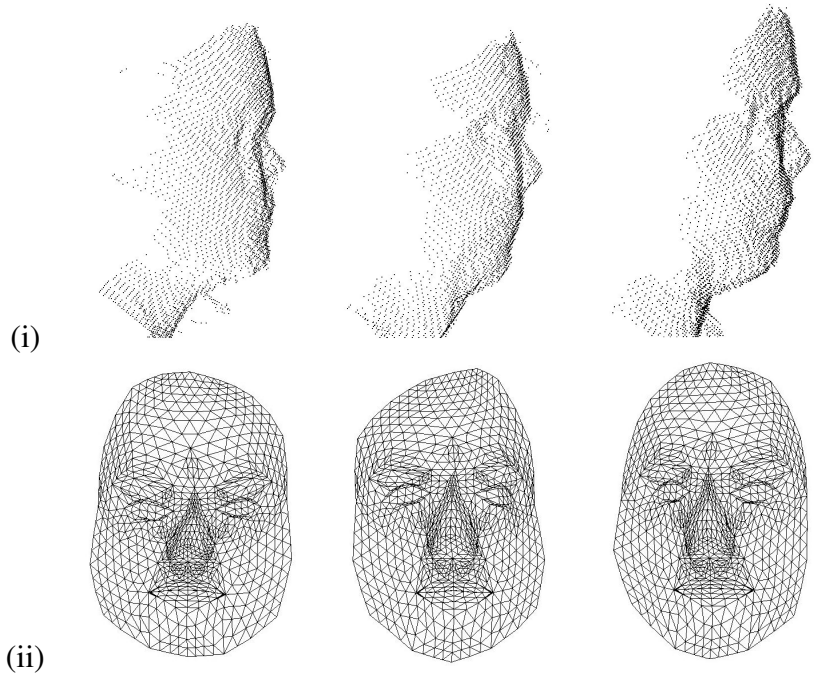
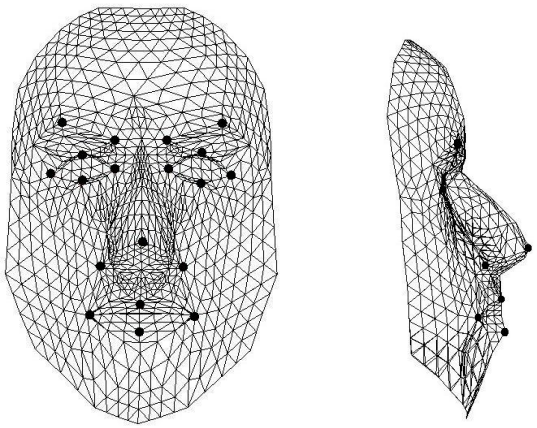
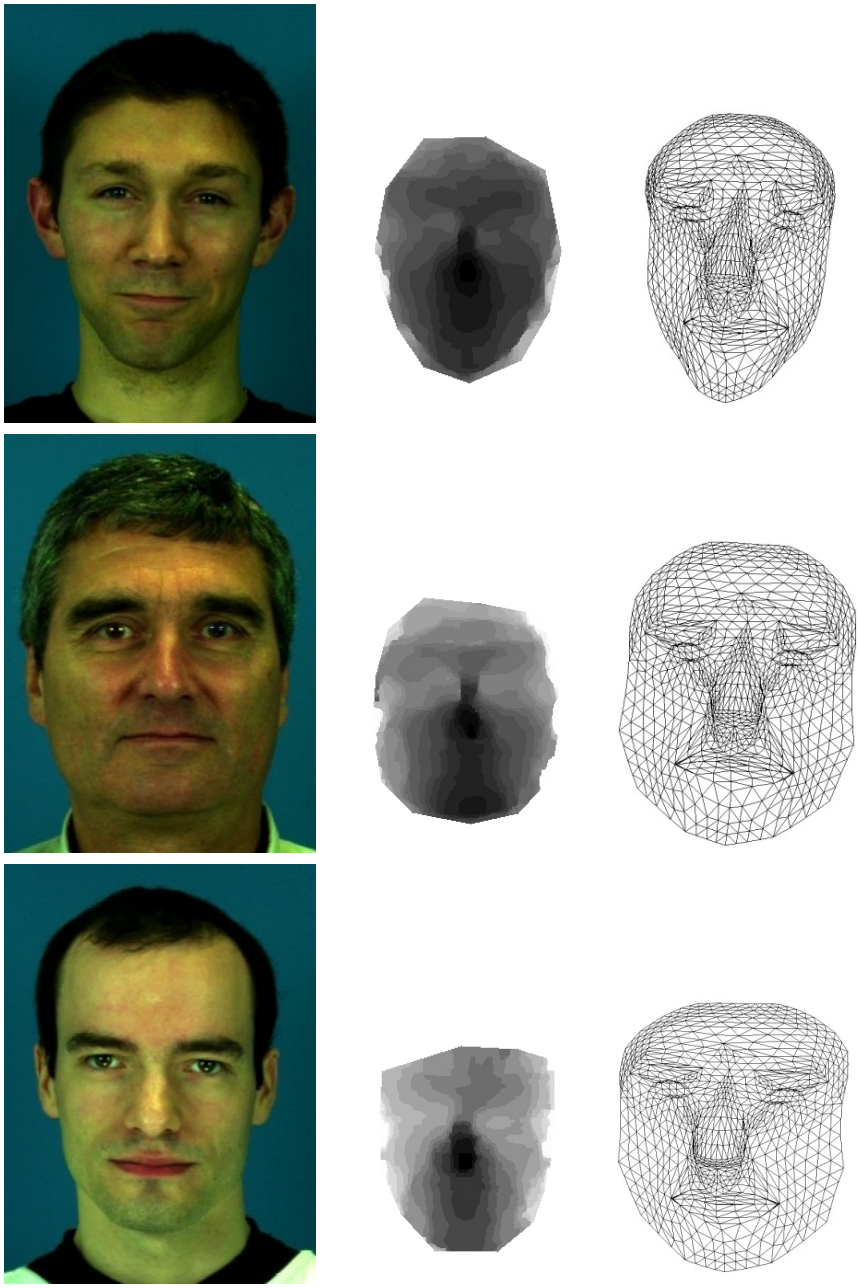


Figure 1.13: Some examples of registration between the candide model and the training set (3D RMA Database [1]): (i) Clouds of points, (ii) Deformations of the Candide model to clouds of points.



(i) (ii)

Figure 1.14: Our Generic Face model, being the mean of the registration of the training set [1] : (i) frontal view, (ii) profile view.



(i) (ii) (iii)

Figure 1.15: Some examples of disparity map and the corresponding registered face model. (i) the face, (ii) the disparity map and (iii) the registered mesh.

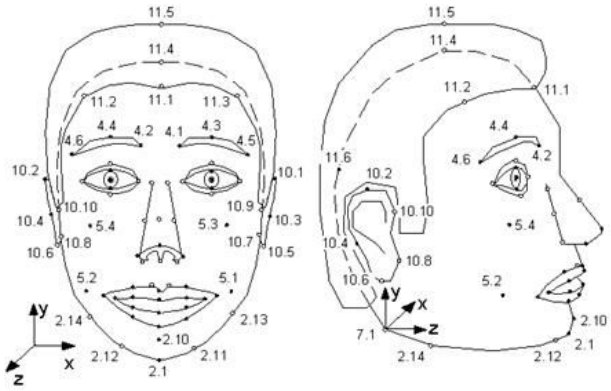
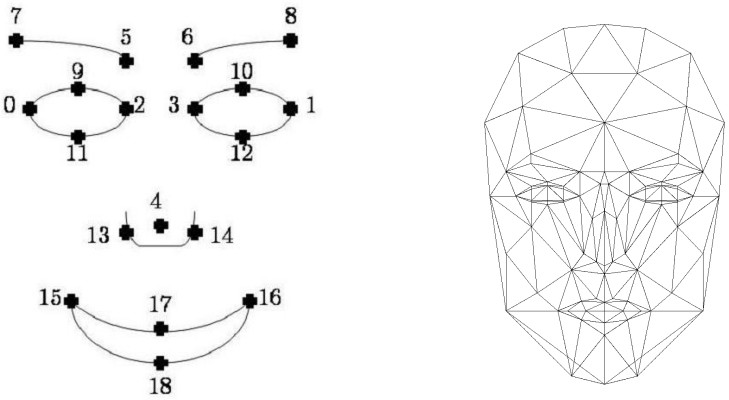


Figure 1.16: Some of the MPEG-4 feature points.



(i) (ii)

Figure 1.17: The definition of our 19 control points guided from the potential representation of expressions and hardware acquisition constraints : (i)in 2D, (ii) on th third Candide Model.

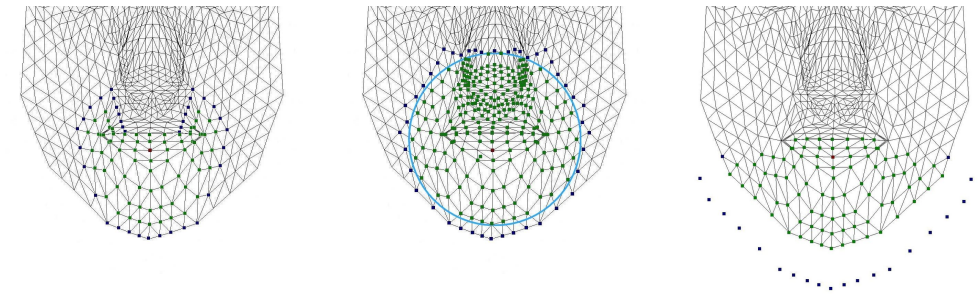


Figure 1.18: An example of animation parameter defined (i) by depth , (ii) with euclidean distance and (iii) manually. In Red, the control point, in blue, the anchor points and in green the point influenced by control point position. In (iii) external control points (for lower lip control point) are visible.

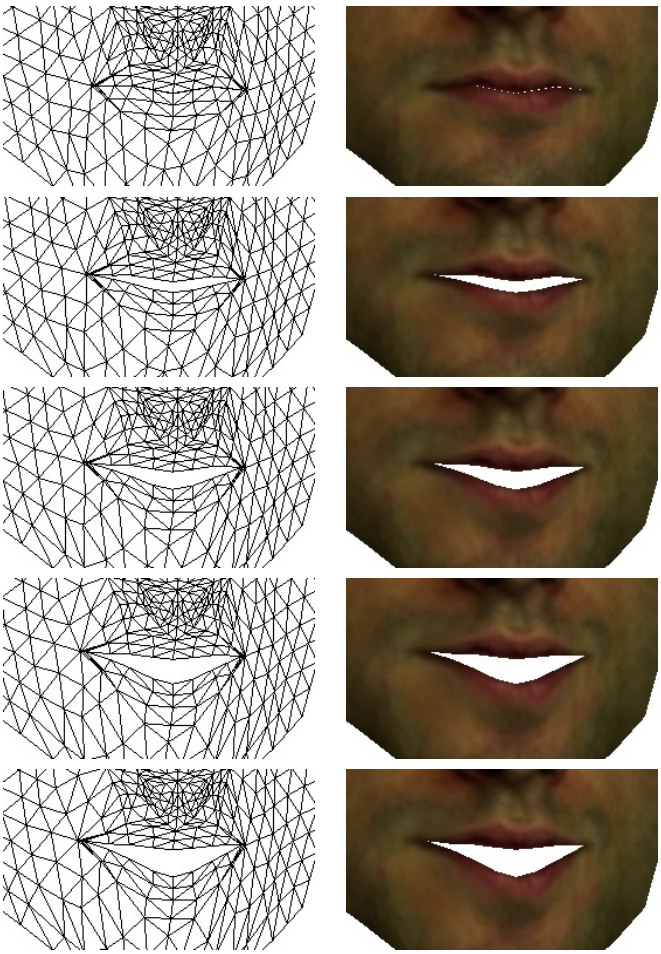


Figure 1.19: Details of the mouth deformation : (i) without texture, (ii) with texture

Chapter 2

Face Inference

Facial pose and movement estimation are critical aspects in Computer Vision and Graphics with applications to various domains, like biometrics, post-production, content creation and manipulation, as well as telecommunications. In this context, and particularly facial behavior analysis, knowing the position of feature is often crucial. Most of the time, facial features refer to points of the face such as corners of the mouth, eyes or eyebrows, tip of the nose or nostrils. Sometimes, in the literature, contours of the eyes, eyebrows of mouth or greyscale patches are also called facial features. Those different approaches of features definitions lead to different detection/extraction techniques:

- Geometric features extraction such as the shape of the facial components (eyes, mouth, etc,...) and the locations of facial fiducial points (corners of the eyes, mouth, etc...).
- Appearance features extraction, representing the texture of the facial skin including wrinkles, bulges and furrows.
- Hybrid methods mixing geometry and appearance based extraction.

2.1 Preliminary work : Face Detection

In most studies on face, such as emotion analysis, reconstruction, recognition, there is a need to know the position of key points on the face. This includes mouth or eyes corners, nostrils, etc .... But before extracting features, it could be interesting to reduce the search space, to speed it up. To this end, most of the features detection techniques require to know where the face is. The most widely used technique used is the one proposed by Viola and Jones. They describe a method for object detection, illustrated in [103] for face detection. The method consists in classifying Haar Basis Functions using Adaboost and adds three main contributions of state of the art :

1. Integral Image : representation of the image which allows to the features to be computed quickly.
2. Fast selection of critical features from a very large number of potential features.
3. Combination of classifiers in cascade.

The results of this method are equivalent to other methods for face detection, but they are remarkably fast. As the face detection uses Haar Basis Functions and Adaboost algorithm, we first introduce those two aspects of the method, before presenting the improvements.

Integral Image

The integral image at point (x, y) is the sum of intensity of pixels (x', y') such that $x' \leqslant x$ and $y' \leqslant y$ or

$$I(x, y) = \sum_{x' \leqslant x, y' \leqslant y} i(x', y') \tag{2.1}$$

where $I(x, y)$ is the integral image at location (x, y) and $i(x, y)$ is the intensity of pixel (x, y). This new representation of the image permits to compute quickly the sum of pixel intensities in any rectangle of an image, as long as we know the integral image value for the four corners. As the method idea is to classify the responses to Haar Basis Functions, once the integral image is computed for the whole image, those response are computed very quickly.

Haar Basis Functions

Haar Basis Functions are so called because they share an intuitive similarity with the Haar wavelets. The 1D version can be defined by $\psi_{jk}(x)$:

$$\psi(x) = \begin{cases} 1 & \leqslant x < \frac{1}{2} \\ -1 & \frac{1}{2} < x \leqslant 1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$\psi_{jk}(x) = \psi(2^j x - k)$$

for j a non-negative integer and $0 \leqslant k \leqslant 2^j - 1$. The 2D version is a very simple extension of the 1D. It is often modeled by a binary mask such as the ones in [Fig.2.1].

Figure 2.1: Examples of Haar Basis Functions. Their neighborhood can vary in height and width.

Adaboost Algorithm

Adaboost is a classification technique based on a set of weak classifiers. It is a linear combination of T weak classifiers defined by $h_t : X \rightarrow \{-1, 1\}$ with error $< 50\%$. Training set is a set of positive and negative patch examples of the object to detect.

$$(\mathbf{x_1}, y_1) \cdots (\mathbf{x_n}, y_n) \text{ where } \mathbf{x_i} \in \mathbf{X} \text{ and } y_i \in Y = \{-1, 1\} \tag{2.2}$$

Each example could be a vector of grey-scales or filter responses (e.g., Haar Basis Functions, Gabor Filter, etc.). For face detection, Viola and Jones use grey-scale patches.. A weight is assigned to each of them indicating their importance in the dataset. At each round t, the best weak classifier and its associated weight are computed, while the weights of incorrectly classified examples are increased. In this manner, the next weak classifier focuses more on those examples. The final classification is given by :

$$H(\mathbf{x}) = sign(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}))), \tag{2.3}$$

The process is presented in algorithm [Algo.1]

As our database contains only frontal view, and as Viola and Jones face detector sources is available, we choose to use them for spending more time in other aspects. (The Viola and Jones implementation is available in the OpenCV library. The Feret Database seems to be used as learning set, but actually no information was given by the developpers.)

Fast selection of critical features

The set of potential weak classifiers for face detection is huge. Considering there are K features (i.e. response to Haar Basis Function) and N examples, there are KN candidates. To reduce that number, the examples are sorted by a given feature value. One can assume that any two thresholds lying between the same pair

Algorithm 1 Adaboost Algorithm according to [103]

Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $yi = 0, 1$ for negative and positive examples respectively.
Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $yi = 0, 1$ respectively, where m and l are the number of negatives and positives examples.
for $t = 1, \ldots, T$ **do**
Normalize the weights, $w_{t,i} =\leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$.
Select the best weak classifier with respect to the weighted error $\epsilon_t = min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$.
Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t , p_t , and θ_t are the minimizers of ϵ_t.
Update the weights: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}$ where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
end for
The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geqslant \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where $\alpha_t = \frac{1}{\beta_t}$

of sorted examples are equivalent. The optimal threshold can be so computed in a single pass over the training set by computing as one goes along the total number of positive and negative examples (considering the weights of each example).

Combination of classifiers in cascade

The main idea of cascade classifiers is to learn how to reject the easily rejectable potential objects in the image, combining successively Adaboost classifiers with a small number of weak classifiers. In the case of face detection, this could be background for example. Then those rejected objects are removed from the learning set and a new training is processed. This mechanism is reiterated until a target of false positive rate is reached. The process of Cascade Adaboost is presented in algorithm [Algo.2].

In the classification process, all candidates feed the first adaboost classifier. It eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing, the number of candidates have been reduced radically, and only the most probable candidates remain (See [Fig.2.3]).

Examples of face detection are presented in [Fig.2.2]. Knowing the position of the face makes the facial features detection easier, by decreasing the search space.

Figure 2.2: Examples of face detection using [103].

2.2 Facial Features Extraction

2.2.1 State of the Art

As already mentioned in the introduction of this chapter, facial features extraction methods can be subdbivided in three categories : appearance based extraction, shape based extraction and hybrid method, mixing appearance and shape.

Texture-based Method

The use of patches to describe features and classifiers to extract them from images has emerged mostly due to progress in machine learning.

Algorithm 2 Cascade Adaboost Algorithm according to Viola and Jones [103]

User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
User selects target overall false positive rate, F_{target}.
P and N are, respectively the sets of positive and negatives examples.
$F_0 = 1.0$, $D_0 = 1.0$ and $i = 0$.
while $F_i > F_{target}$ **do**
  $i \leftarrow i + 1$.
  $n_i = 0$ and $F_i = F_{i-1}$.

  while $F_i > f \times F_{i-1}$ **do**
    $n_i \leftarrow n_{i+1}$.
    Use P and N to train a classifier with n_i features using AdaBoost.
    Evaluate current cascaded classifier on validation set to determine F_i and D_i.
    Decrease threshold for the i^{th} classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$ (this also affects F_i).
    $N \leftarrow \emptyset$

    if $F_i > F_{target}$ **then**
      Evaluate the current cascaded detector on the set of non-face images and put any false dectections into the set N.
    end if
  end while
end while
The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geqslant \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$

where $\alpha_t = \frac{1}{\beta_t}$

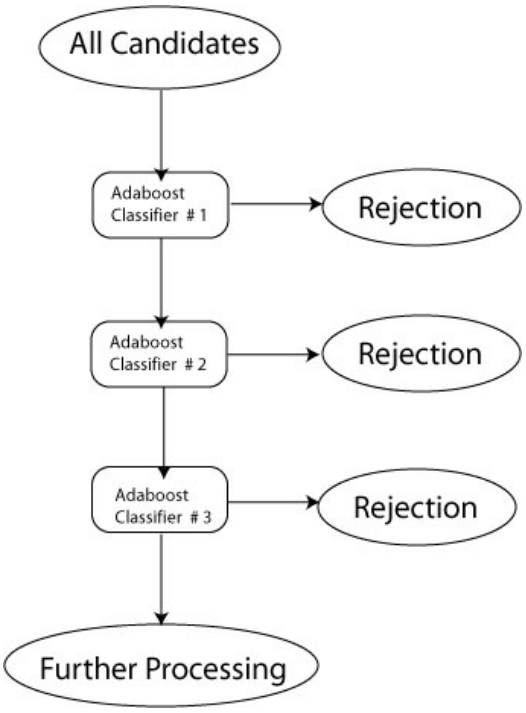


Figure 2.3: Schema of the Cascade Adaboost Process. All candidates feed the first adaboost classifier and are little by little rejected to only keep some of them.

In [104] facial interest points are determined through training Gentleboost algorithm [41] of which responses describe the similarity between the object to be classified and the object learned during the training phase. Gentleboost is used in the cascade algorithm of Viola and Jones [103] for object detection, instead of Adaboost. It is trained on the Gabor filter responses of positive and negative examples of features. Once the face is detected, the following step consist in dividing the face area in 20 region of interest, corresponding to the 20 features to detect. To help in defining those regions for the eyes and the mouth, vertical and horizontal histograms are computed in upper face area, the iris coordinates are detected as the coordinates corresponding to peaks in the histogram. Using the distance between the iris, they define approximately the region of the mouth. To detect the medial point of the mouth, the y-coordinate is computed using the same histogram method like for the eyes (as the region between the two lips is the darker one). The x-coordinate correspond to the middle point between the eyes. For each point of the 20 ROI, one 13×13 pixel patch is extracted in the gray scale image, and 48 are extracted by the 48 representations of the Gabor filtered ROI (8 orientations and 6 spatial frequencies). GentleBoost reduces the dimensionality of the feature representation by removing redundant information and becomes faster than Adaboost. The authors achieve a mean detection rate of 93%.

In [19], Chen et al. constrain a probabilistic-like output map with a 3D model, to recover position of nose, eyes and the corners of the mouth. From a database of positive and negative examples, characteristics of features are computed using a unified Boosting Chain Learning algorithm [113]. This technique permits to produce a single probabilistic output, instead of series of output as Cascade Adaboost does. But as the output map is noisy, particularly around features position, the authors model it using a gaussian mixture model. Thus the estimation of the gaussians gives the features position. All possible configurations of the 5 features are then refined by a 3D shape constraint, to select the best one. But the face model used here, is really simple and considers the eyes and the mouth in the same plane. This method could remove the ambiguity of incoherent points, but one can be doubtful concerning two neighbor candidates for the same feature and the detection precision.

Duffner and Garcia, [31] propose a connexionist approach to produce feature maps (where the intensity is high where the feature is detected). The neural network is composed of 6 layers, including input and output layers. Actually, the first four layers transform the input image by convolution with kernels, pushing foreward the oriented edges or the corners, and sub-sampling. The goal is to make the system less sensitive to small shifts, distortions, scale or rotations. The two last layers permit to produce the 4 feature maps (one for each of the feaure) from the output of the 4^{th} layer. In [36], Feris et al. use also a connexionist approach,

but that can be considered closer to Active Appearance Model [22], as, in a hierarchical approach, they try to 'reconstruct' first the appearance of the face, and then, the features appearance. Gabor Wavelet Network (GWN) are used for the reconstruction, represented by $\Psi(\mathbf{x}) = \sum_{i=1}^{n} w_i \psi_{n_i}(\mathbf{x})$ with ψn_i is one of the n wavelets, w_i is the associated weight, computed by gradient descent and n_i is a set of parameters coding the scaling, orientation and translation. One can compare this representation as a compression representation in the same way as Principal Component Analysis. Such a GWN is first used for the face detection, and then for the feature localization.

In [47], the authors present a topological method to recover eyes centers and mouth center. Even if the method is assumed to be topological, they don't directly use the shape of features. The face is converted in a 3D surface where each pixel (x, y) of intensity $I(x, y)$ corresponds to a 3D surface point $(x, y, I(x, y))$. Assuming this face representation, eyes and mouth are composed of crests and valleys, and particularly, borders of the eyes, irises and the separation of the lips are ravines (as their intensity is usually very low). Ravines are defined as points of a surface here the maximum curvature is a local maximum in the corresponding principal direction. Eyes are first detected, together, considering they are symmetric and have the same size and intensity. Finally, the eyes permit to help in the mouth localization using the inter-lips ravin.

The main limitation of such methods relates with the fact that, for most of them, features are detected individually without taking into account the expected facial geometry, such that approximative symmetry of the face or relative position of features regarding to each other.

Shape-based Method

Global face models is a natural approach to introduce anthropometric constraints. Active Shape Models [24, 67], usually called ASM, are statistical models coding the shape of objects and fitting an example of a new object by iterative deformation. The shape is defined by a vector X of a set of n landmarks, and his variations, extracted from labeled examples by Principal Component Analysis. Thus a new shape X can be defined by $X \approx \bar{X} + Pb$. $\bar{X}$ is the mean of all the shape in the database, P is a matrix of the first t eigenvectors of the database covariance matrix, and b is the vector of model parameters. The variance of the i^{th} parameter P_i across the training set is given by the eigenvalue λ_i (b_i is limited by $-3\lambda_i$ and $3\lambda_i$). To transform the shape model in a target shape, a first global Euclidean transformation (translation, rotation and scale) is typically used. Then, recovering the local structure is an iterative process. After the initialization by

the mean shape, the first derivatives in the region of each landmark is compared to the statistical model defined in the learning step, to find the best match (along the profile line for the edge locations). The transformation parameters are then updated. This process is then repeated until there is no significant change in the shape parameters.

Active Shape Models are the basis for a lot of studies, and particularly for facial feature extraction. Mahoor et al. [75] are ones of those who try to enhance the technique. They propose three improvements ; First, as ASM are very sensitive to the initialization, they use the raw estimation of eyes and mouth position [52] as a first approximation of the face position. Then, the statistical model coding each of the landmark is not only in grey level, but the author use independently the three color channels, i.e. Red, Green and Blue. Finally, they improve features location around the mouth by segmenting the region in two categories : lips or skin. This is done using the Fisher Discriminant Analysis in RGB color space. Features extraction method proposed by Zuo and de Width [124] enhances also the classical Active Shape Methods but in a different manner. The initial position of the model is defined by the best match between the average face template and the gradient map of the image, up to scale. The main improvement concerns the landmarks model. Where [24, 67] and [75] use a statistical model, Zuo and de Width propose to use Haar-wavelet representations of local texture patches, containing more information than a simple contour, and are nevertheless fast to compute, according to the integral image technique [103].

Hennecke et al. [50] propose to use a deformable model for mouth extraction. This method is not based on statistics from a database, but on the registration of a template according to image data. The template is defined by 4 curves (parabolas and quartics), around a center C and an inclination θ. Those parameters are updated minimizing a cost function E. E consists in four curve integrals (one for each curve), based on the vertical gradient of the image (as the mouth edges are horizontal in majority). The authors add spatial and temporal constraints. The spatial constraints make sure the template shape stay reasonable, while the temporal constraint add spring forces to keep lip thickness close to the mean when tracking the mouth.

In these methods, authors focus only on shape or texture for facial features extracation. Hybrid method take advantages of both mixing the shape and the texture of facial point of interest to detect them.

Hybrid Method

Active Apperance Model (AAM) [22] is a statistical model to represent objects shape and texture. The model is built during a training phase from a set of annotated texture images (the case of faces is particularly used to illustrate AAM). The mean and the matrices depicting the modes of variation of shape and texture, are computed and by varying the parameters of the model, any face can be expressed using a linear combination of the modes of variation.

The AAM algorithm uses the texture residual between the model and the target image to rebuild the face. It has been heavily considered in this context [26]. However the authors propose a technique where, a selection of the nearest neighbor in the learning database is made, according to the current shape and texture. The AAM parameters are then updated. This is reiterated until convergence. It requires the ability to reconstruct any combination of patches through a weighted sum of parameters and so requires too a huge quantity of data. Such approaches have been often considered for faces. The main limitation of this a method is the ability to describe inter and intra-user variations with a linear subspace. The underlying statistical assumption is hard to be satisfied while at the same time recovering the model parameters towards capturing all possible deformations and different subjects is rather challenging.

In [112], the authors propose a method combining 2D Active Appearance Models (AAMs) and 3D morphable models (3DMMs) [11], by fitting combined 2D+3D active appearance models, for face tracking. This method could give, in the same way, the position of features in 3D. But as in [26], this requires to be able to synthesize any face. It seems possible in theory, but is difficult to reach due to the high faces variance. Morphable face models [11] is a more elegant approach to face representation and tracking. They consist of a rather dense triangulated model that can involve texture, shading, etc,... The inference process is done through the deformation of the triangulated surface such that the synthesized image matches the observed one. However such an approach is expansive from computational point of view.

The method in [26] is based on AAM. For each image in the database, a rectangular patch is stored with the corresponding shape model b for each feature point. The algorithm begins by an initialization of the shape model. A set of feature patches are generated using a nearest neighbor approach and parameters b are computed according to the initialization. These parameters are compared to training data. The K closest shape in the training are selected and their associated patches are compared, using normalized correlation to the feature templates.

The best matching training example textures are then used to form detectors for each facial feature. The next step is a refinement step. The feature detectors are applied to the current image, to compute response images. To each feature point i, with coordinates (X_i, Y_i), a response $I_i(X_i, Y_i)$ is computed. The vector $X = (X_1, \ldots, X_i, \ldots, Y_1, \ldots, Y_i, \ldots)$, in the same time, is computed from the shape parameters b, and a similarity transformation T_t. T_t and b are concatenated into p. So X can be expressed as a function of p, and the refinement can be performed by optimizing a function

$$f(p) = \sum_{i=1}^{n} I_i(X_i, Y_i) + R \sum_{j=1}^{s} \frac{-b_j^2}{\lambda_j}$$

(the second term being an estimate of the log-likelihood of the shape given shape parameters b_j and eigenvalues λ_j). The process is reiterated until points converged.

The method proposed in [53] follows a linear combination model. The method uses the knowledge of prototypic faces to interpret novel faces. During the training setp, the prototypes are labeled manually and a 5×5 matrix is used to depict the gray-value neighborhood of each labeled point. The following step is the alignment of the prototypic faces. A bootstrapping algorithm is proposed by Vetter et al. [7] for the alignment. This uses optical flow algorithm to align the labeled faces, without taking into account the labels. The idea proposed by the authors is to restrained the optical flow algorithm to keep the correspondences of the labeled points fixed. Once the alignment is done, the linear combination model has to be constructed. Considering all the prototypic faces, the image I_0 is considered as the reference. The correspondence between I_0 and all the others images is denoted as $S_i : R^2 \rightarrow R^2$ such that the shape free image I_i is defined by $T_i(x, y) = I_i(S_i(x, y))$ where (x, y) is a point in I_0. T_i is regarded as the image which has the shape of the reference face I_0 and the texture of I_i. A face can be expressed as :

$$I^{com} = (A \circ \sum_{i=1}^{N} a_i S_i) = \sum_{i=1}^{N} b_i T_i$$

where $A \circ \sum_{i=1}^{N} a_i S_i$ and $\sum_{i=1}^{N} b_i T_i$ are the combination face's shape and texture and A is an affine transformation. Principal Component Analysis technique is then used to compress the data, as information in the model is redundent.
The matching between the target face image and the combination model is achieved using methods such as stochastic gradient descent algorithm, to minimize the energy :

$$E = \sum_{x,y} [I^{input}(A \circ S^{com}(x,y)) - T^{com}(x,y)].$$

The last step consists in the facial feature extraction. The result of the matching processinf is a combination of parameters fitting the input face. The feature points of the input face can so be extracted. To get a more precise estimation, the positions are refined using the gray feature matrices. For each estimated feature points, a neighborhood is given, denoted as the matching spaces. For each point of this space, gray feature matrices are computed. Then, the same matrices are computed for the computed combination model. The accurate feature points are then defined by the position, in the neighborhood where the combination model matrices give the best matches.

In [25] the author proposed a classical facial feature extraction technique using AdaBoost, with shape constraints. The Viola-Jones face detector is used to detect the face in the image. Then, they use Adaboost algorithm to extract the facial features. The sets of candidate feature points, selected by Adaboost, are tested using shape constraints in two ways. Firstly a shape model is fitted to the set of points and the likelihood of the shape assessed. Secondly, limits are set on the orientation, scale and position of a set of candidate feature points relative to the orientation, scale and position implied by the global face detector. But, this method is limited by the number of features points and the number of candidates for each of them, otherwise, the computation time explodes. The number of features points have to be limited (4 in the paper) and it is the same for candidate number (5 in the paper). The authors add an efficient search method to select the highest scoring candidate with the maximum number of feature points that satisfy the shape constraint.

2.3 Anthropometric constraints for facial features extraction

Texture based approaches prove their efficiency in the context of facial features extraction. But face artifacts, beauty spot or wrinkles, could lead to detection error. Furthermore, texture based approaches could give several candidates for each feature points. A selection have to be made among all candidates. The introduction of anthropometric constraints refers most of time in Active Appearance Models. But AAM are not enough generic. As the goal is to reconstruct the face, the learning database should contain all existing facial appearance. As face's appearance and geometry are unique, such a database cannot exist. Anthropometric

constraints combined to texture based extraction method, such as adaboost, permits to select the best feature points among all candidates and could enhance the detection [46].

We assume a rather standard point representation, according to the face model that we presented in the previous chapter, and then seek to optimize their position. We introduce a novel way to impose global consistency through a number of local interactions between control points. Let us assume that n responses, trained from positive and negative examples, are available for each control point (if the number is lower, then we can replicate entities towards recovering the same number). Then, we can define a label set with potential correspondences for each control point and the task of finding the optimal configuration can be viewed as a discrete optimization problem.

In such a context, one should recover a configuration that refers to strong responses of the Adaboost classifier, while at the same time it is consistent with the expected face geometric characteristics. This allows to define features position using a Markov Random Field formulation. The objective function consists of pair-wise potentials that account for the admissible set of positions between two points, while the singleton potentials correspond to the score of classifiers given the observation and the control point under consideration. The resulting Markov Random Field involves pair-wise potentials that are not sub-modular functions and therefore, we use an efficient technique from linear programming [65] towards recovering its lowest potential.

2.3.1 Markov Random Filed formulation of the problem

Let us consider a discrete set of labels $\mathcal{L} = \{l^1, ..., l^P\}$, with P the number of labels, corresponding to the potential image candidates for the control points $\mathcal{D}_{\mathbf{m}=\{\mathbf{x}_0^{l_1},...,\mathbf{x}_0^{l_N}}, ..., \mathbf{x}_N^{l_1}, ..., \mathbf{x}_N^{l_P}\}$, N being the number of control points. A label assignment l_p to a grid node associated with the model control point n and the image coordinate $\mathbf{x}_n^{l_p}$. Let us also consider a graph $\mathcal{G}$ which has, as nodes, the $N = 19$ control points of the model.

One can reformulate the optimal correspondence selection problem as a discrete optimization problem, where the goal is to assign individual labels $l0$ to the grid nodes. In this context, the image support term $E_{im}(\cdot)$ can be rewritten as:

$$E_{im}(\theta) = \sum_{n \in \mathcal{G}} \underbrace{g\left(\sum_{t=1}^{T} \alpha_t h_t\left(\mathbf{x}^{l_n}\right)\right)}_{\approx V_n(a_{\mathbf{m}})}, \tag{2.4}$$

with g being a monotonically decreasing function. Computing Adaboost response for all pixels in the image and all features could be a waste of time, when some

pixels, like background pixels are obviously not potential candidates. We first extract candidates using Cascade Adaboost algorithm, to detect candidates and use the classic adaboost algorithm to compute the image term.

Let's remain briefly the Adaboost algorithm. Adaboost is a linear combination of T weak classifiers defined by

$$h_t : X \rightarrow \{-1, 1\}$$

with error $< 50\%$. Training data is a set of positive and negative patch examples of the object to detect $(\mathbf{x_1}, y_1) \cdots (\mathbf{x_N}, y_N)$ where $\mathbf{x_i} \in \mathbf{X}$ and $y_i \in Y = \{-1, 1\}$. Each example could be a vector of grey-scales or filter responses (e.g., Haar Basis Functions, Gabor Filter, etc). A weight is assigned to each of them indicating their importance in the dataset. At each round t, the best weak classifier and its associated weight are computed, while the weights of incorrectly classified examples are increased. In this manner, the next weak classifier focuses more on those examples. The classification is given by:

$$H(\mathbf{x}) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right), \tag{2.5}$$

As our total system is composed of several step, any time gain is important. In [104] the data are the gray level of patches and their response to Gabor filters. In [103], where the object to detect is the face, they use the integral image. Even if they propose a method to speed up the computation, the initialization process could be quite long without special computation tricks or GPU computation. So we choose to work with 15×15 patches, in gray level, whose histograms are equalized and their responses to the Haar Basis Functions shown in [Fig.(2.4)]. The size of these patches was chosen according to different cross-validation test done on our training set (our six basic emotions corpus, including not only neutral faces but also expressing faces) for patch size from 9×9 pixels to 19×19 pixels. The best results were obtain with 15×15 pixels.

The next aspect to be addressed is the introduction of proper interactions between label assignments. It can be done through the use of anthropometric constraints $E_{an}(\cdot)$ in the label domain.

Indeed the most obvious constraints is the face symmetry. Even if a face is not exactly symmetric, considering the 6 basic emotions [Fig.3.6], the movements of feature points are rather symmetric with certain errors that are not critical. We assume that when expressing joy, if the left corner of the mouth goes up and the right corner goes up too, with the same intensity. Examples of symmetry correspondences and inequality are presented in [Tab. (2.1)] and detailed in Appendix

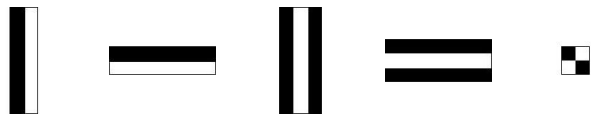


Figure 2.4: The 5 Haar Basis Functions used for Adaboost Classification of size (i) 4×15 pixels, (ii) 15×4 pixels, (iii) 6×15 pixels, (iv) 15×6 pixels, (v) 4×4 pixels.

$(\mathbf{m}, \mathbf{n})$	Description	Constraints	$V_{\mathbf{m},\mathbf{n}}$
(15,16)	Corners of the lips	$Symmetry$	$(15.y - 16.y)^2$
(5,6)	Inner corners of the eyebrows	$Symmetry$	$(5.y - 6.y)^2$
(9,10)	Midpoints of upper eyelids	$x = \frac{0.x+2.x}{2}$	$(9.y - 10.y)^2$ and $(9.x - \frac{0.x+2.x}{2})^2$
(11,12)	Midpoints of lower eyelids	$x = \frac{0.x+2.x}{2}$	$(11.y - 12.y)^2$ and $(11.x - \frac{0.x+2.x}{2})^2$

Table 2.1: MPEG-4 anthropometric constraints and corresponding pair-wise potentials. All of the anthropometric constraints are summarized in the Appendix A.

A. These constraints can be formulated as follows:

$$E_{an}(\theta) = \sum_{m \in \mathcal{G}} \sum_{n \in \mathcal{N}(m)} V_{mn}(l_m, l_n), \tag{2.6}$$

where $\mathcal{N}$ represents the neighborhood system associated with the deformation grid $\mathcal{G}$ and $V_{mn}(l_m, l_n)$ represents the pairwise potentials lonking points m and n according, rexpectively, to the labels l_m and l_n. Towards addressing computational constraints, we consider only pair-wise relations between control points of the model : first corner points such as the corner of the eyes or of the mouth, nostrils or eyebrows. Then, once they are detected, we add the middle points on the eyelids or on the lips and the tip of the nose, taking into account their x-coordinates is enforced by the previously detected points.

2.3.2 Optimization process through Fast-PD

For optimizing the above discrete Markov Random Field, we will make use of a recently proposed method, called Fast-PD [65]. This is an optimization technique, which builds upon principles drawn from the duality theory of linear programming in order to efficiently derive almost optimal solutions for a very wide class of NP-hard MRFs. Instead of working directly with the discrete MRF optimization problem above, Fast-PD first reformulates that problem as an integer linear programming problem (the primal problem) and also takes the dual of the corresponding LP relaxation. Given these two problems, i.e. the primal and the dual, Fast-PD then generates a sequence of integral feasible primal solutions, as well as a sequence of dual feasible solutions. These two sequences of solutions make local improvements to each other until the primal-dual gap (i.e. the gap between the objective function of the primal and the objective function of the dual) becomes small enough. Once this happens, the last generated primal solution is guaranteed to be an approximately optimal solution, i.e. within a certain distance from the optimum (in fact, this distance can be shown to be smaller than the achieved primal-dual gap). This is exactly what the next theorem, also known as the Primal-Dual Principle, states.

Primal-Dual Principle 1 (Primal-Dual principle) *Consider the following pair of primal and dual linear programs:*

$$\begin{array}{llll} \text{PRIMAL:} & \min \mathbf{c}^T\mathbf{x} & \text{DUAL:} & \max \mathbf{b}^T\mathbf{y} \\ & \textit{s.t.} \quad \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geqslant \mathbf{0} & & \textit{s.t.} \quad \mathbf{A}^T\mathbf{y} \leqslant \mathbf{c} \end{array}$$

and let $\mathbf{x}, \mathbf{y}$ *be integral-primal and dual feasible solutions, having a primal-dual gap less than* f*, i.e.:*

$$\mathbf{c^T x} \leqslant f \cdot \mathbf{b^T y}.$$

Then $\mathbf{x}$ *is guaranteed to be an* f*-approximation to the optimal integral solution* x^**, i.e.,* $\mathbf{c^T x^*} \leqslant \mathbf{c^T x} \leqslant f \cdot \mathbf{c^T x^*}$

Fast-PD is a very general MRF optimization method, which can handle a very wide class of MRFs. Essentially, it only requires that the pairwise potential function is nonnegative (i.e., $V_{pq}(\cdot,\cdot) \geqslant 0$). Furthermore, it can guarantee that the generated solution is always within a worst-case bound from the optimum. In fact, besides this worst-case bound, it can also provide per-instance approximation bounds, which prove to be much tighter, i.e. very close to 1, in practice. It thus allows the global optimum to be found up to a user/application bound. Finally, it provides great computational efficiency, since it is typically 3-9 times faster than any other MRF optimization technique with guaranteed optimality properties.

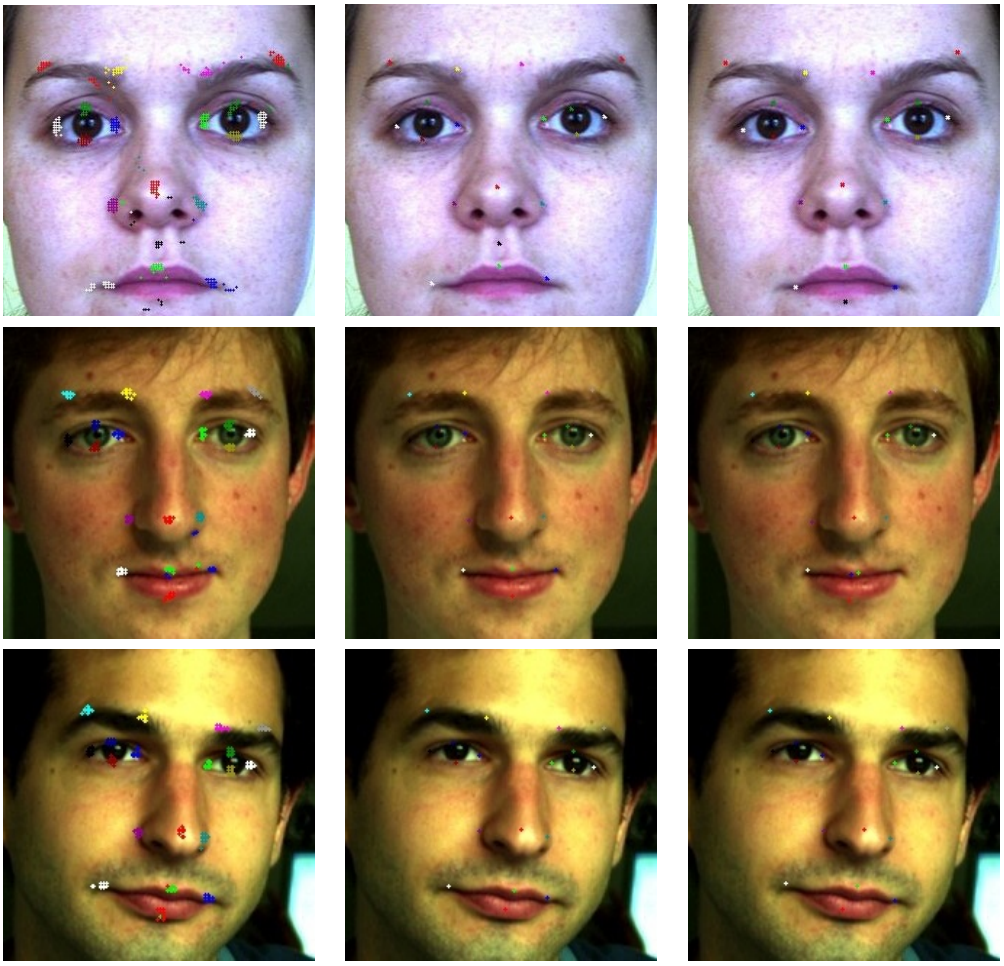
[Fig.(2.5)] presents results of the facial features extraction method, presented here. (i) shows the 20 candidates for each feature point, while (ii) presents the selection of feature points with the highest score and (iii), the optimal configuration defined by the anthropometric constrained and optimized by Fast-PD.

2.4 3D Pose Estimation from a single image

Facial features extraction gives the feature position in 2D. Pose estimation in 3D from 2D images is a more challenging problem, in particular when aiming capturing non-rigid behavior. We propose a novel class of prior models that encode the position of a control point. The problem is casted using a conditional Markov Random Field, comparable to the ones used in facial features extraction but involving different prior term. It imposes/guarantees that the 3D model to be recovered is part of the set of admissible 3D faces, while the data term aims to establish correspondences between the image and the model. Such correspondences are obtained by maximizing the response of an Adaboost classification procedure of the model projections from the 3D configuration on the image.

2.4.1 The Prior Constraints

Let us consider a set of 3D points $\mathcal{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N, \}$ be a set of points or landmarks. The task, of modeling shape variations consists of recovering a distribution for the same set of control points among examples of a training set. Prior to the construction of such representation, one should induce invariants to the model. Let us consider a set of examples $\{\mathcal{S}_1, ..., \mathcal{S}_m\}$, and $\mathbf{x}_n^k, (n,k) \in [1,N] \times [1,k]$ being the position of the i-th control point of the example k. Introducing a model in such a space requires registration between samples, that is a problematic and tedious task. Without loss of generality, let us assume that for the time being in



(i) (ii) (iii)

Figure 2.5: Constrained Extractions : (i) The candidates, (ii) The highest score configuration, (iii) The optimal configuration defined by Fast-PD.

the training set, we have mostly translation, rotation and some local variations. Then, one can consider for a given example k the norm of the euclidean distance between all pairs points, that is:

$$d(n,m) = |\mathbf{x}_n^k - \mathbf{x}_m^k|,\ (i,j) \in [1,N] \times [1,M] \tag{2.7}$$

Such representation is invariant to global translation and global rotation. This is not the case for scale variations. Let us now consider the case of a dense parameterization of the surface and a scale factor of s. Then, it is trivial to prove that, for any pair of points, $d(n,m|s) = sd(n,m)$. One can now consider the norm of distance normalized using the sum of all distances for a given example k, that is

$$\hat{d}(n,m) = \frac{|\mathbf{x}_n^k - \mathbf{x}_m^k|}{\sum_{u=1}^{N-1}\sum_{v=u+1}^{N}|\mathbf{x}_u^k - \mathbf{x}_v^k|} \tag{2.8}$$

and in such a context it is straightforward to prove invariance to translation, rotation and scale.

Once such representation has been considered, the next step consists of modeling shape variations. We would like to determine a prior density $p(\mathbf{x}_1, ..., \mathbf{x}_N)$ for the set of points $(\mathbf{x}_1, ..., \mathbf{x}_N)$. If we assume that the position of each point can be defined from the position of the other control points, one can write that

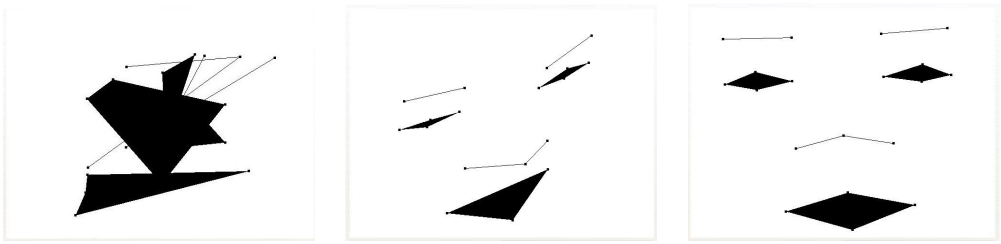
$$p(\mathbf{x}_1, ..., \mathbf{x}_N) \approx \prod_{n=1}^{N} p(\mathbf{x}_n|\mathbf{x}_1, ..., \mathbf{x}_{n-1}, \mathbf{x}_{n+1}, \mathbf{x}_N) \tag{2.9}$$

where one can note some abuse on the notation for the case of $i = 1$ and $i = N$. Such a model assumes that the position of a control point is conditional to the position of the rest of the volume, and at individual level, encodes local structure. Given that, we would like to impose invariance with respect to similarity transformations, one can introduce the shape manifold as:

$$p(\mathcal{S} : \mathbf{x}_1, \ldots, \mathbf{x}_N) \approx \prod_{n=1}^{N} p(\mathbf{x}_n|\hat{d}(n,1), \ldots, \hat{d}(n,n-1), \hat{d}(n,n+1), \hat{d}(i,N)) \tag{2.10}$$

We can go even further and assume pairs of points being independent which will lead to a less precise approximation of the density being sufficient though if the number of dependencies between control points is significant, leading to:

$$p(\mathcal{S} : \mathbf{x}_1, ..., \mathbf{x}_N) \approx \prod_{n=1}^{N}\prod_{m=1, n\neq m}^{N} p(\mathbf{x}_i|\hat{d}(n,m)) \tag{2.11}$$



(i) (ii) (iii)

Figure 2.6: The application of the prior knowledge based constraints : (i) the random 19 features, (ii) New position of the 19 features after applying the prior constraints, (iii) the frontal view.

The use of such model encodes the global aspect of human face, while being expressed as local combinations of individual densities. As a proof, [Fig.(2.6)] shows 19 points randomly defined in the 3D space. When applying such a model, the structure is deformed such as the features 3D position satisfy the pair-wise constraints and describe a human face. Furthermore, such a model can account for local variations of varying complexity due to the fact that different statistical models can be used to express the pair-wise densities towards capturing the observed variation. In the context of our approach, we have used simple Gaussian densities to express pair-wise interactions models between the control points.

In addition to learn constraints from the relative position of points, the case of facial modeling can inherit knowledge from the face anatomy. Examples include symmetry, equal-distance between pair of points, etc. Those constraints have already been addressed for facial features extraction. One can see examples of such constraints in [Tab. (2.1)] while a complete description is given in Appendix A.

2.4.2 The Pose Estimation

Let us now consider a new image, where the pose of the face should be determined. Given the prior constraints being defined previously, the most appropriate solution will be to deform the 3D model such that the projection of the features on the image corrseponds to their position on the face. Let us consider that such an optimization term involves a shape prior term and an image one:

$$E(\mathcal{S}) = \sum_{n=1}^{N} f(S_n(\pi(x_n))) + \alpha \sum_{n=1}^{N} \sum_{m=1, m\neq n}^{N} -\log p(x_n|\hat{d}(n,m)) \qquad (2.12)$$

where $S_n()$ is a fitting function giving a similarity coefficient of a point according to the feature point n, $\pi(x_n)$ is the projection of the 3D features $\mathbf{x}_i$ in

the image plane (projection matrix being known), f is a function inversely proportional to the fitness between the projection on the image and expected feature properties and α a weighted constant.

Shape-driven Pose Estimation

Let us consider a set $\mathcal{L}$ of labels $l^1, ..., l^P$ for each feature, associated with a set of discrete 3D displacements vectors $\mathcal{D} = d^1, \ldots, d^P$. Each of these displacements in x, y and z directions can refer to a new position on each nodes of a 3D grid. (See [Fig.(2.8)]). One can view the deformation of our model equivalent to the deformation of the control points of a reference shape $\mathcal{S}_{ref}$ according to a set of displacements, and the set of labels l that minimizes the following function:

$$E(l|\mathcal{S}_{ref}) = \sum_{n=1}^{N} \sum_{m=1, m\neq n}^{N} -\log\, p(\mathbf{x}_n + d^{l_n} | \hat{d}(n,m)) \tag{2.13}$$

Image-driven Pose Estimation

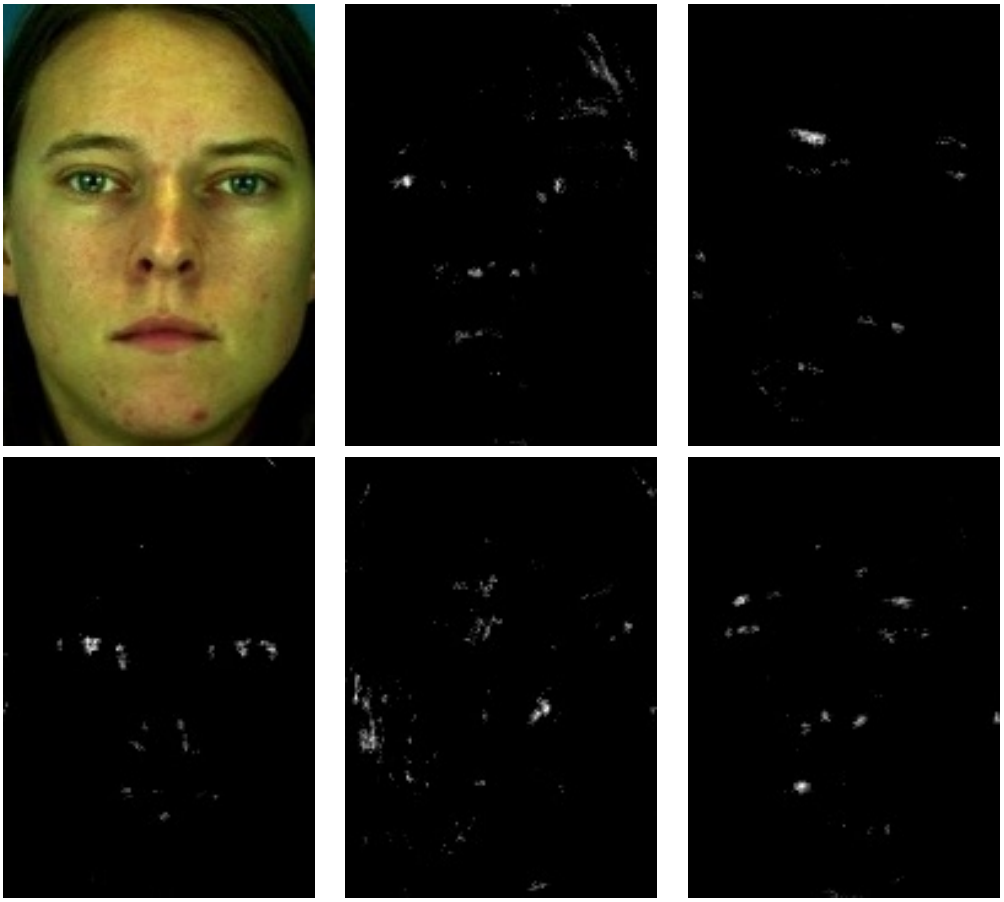
Considering N^3 3D candidates, each of them being a node of a 3D grid and projected in the image space using the projection matrix P: $m = P \times M$ where M is the 3D candidate and m is its projection on the image (see [Fig. (2.8)] for an example) . Then the data cost is computed as inversely proportional to the weighted sum of weak classifiers, whose the sign is used in Adaboost algorithm to classify data. The response, more than the sign, give us an idea on the possibility for a point to be or not what we are looking for, regarding another candidate. The highest the response is, the most probable the point is to be the feature. [Fig. (2.7)] shows some of the Adaboost response map of different features.

However, we have to pay attention of the fact that a projection does not lie necessarily exactly on the feature place, and so can be rejected by Adaboost, and be, in the same time, the closest candidate. So we choose to use the highest Adaboost response in a neighborhood. This leads to a final score such as:

$$S(\mathbf{x}) = \underset{\mathbf{y}\in\mathcal{V}(\mathbf{x})}{\arg\max} \sum_{t=1}^{T} \alpha_t h_t(\mathbf{y}), \tag{2.14}$$

where, $\mathcal{V}(\mathbf{x})$ is the neighborhood of $\mathbf{x}$.

In such a context, the problem of finding the most appropriate deformation of a reference shape can be expressed using an MRF with singleton and pair-wise



(i) (ii) (iii)

Figure 2.7: Adaboost response map of different features. On the first line : (i) Original Image, (ii) Left eye outer corner, (iii) left eyebrow inner corner. On the second line (i) Left upper eyelid middle point, (ii) Right nostril, (iii) Left mouth corner.

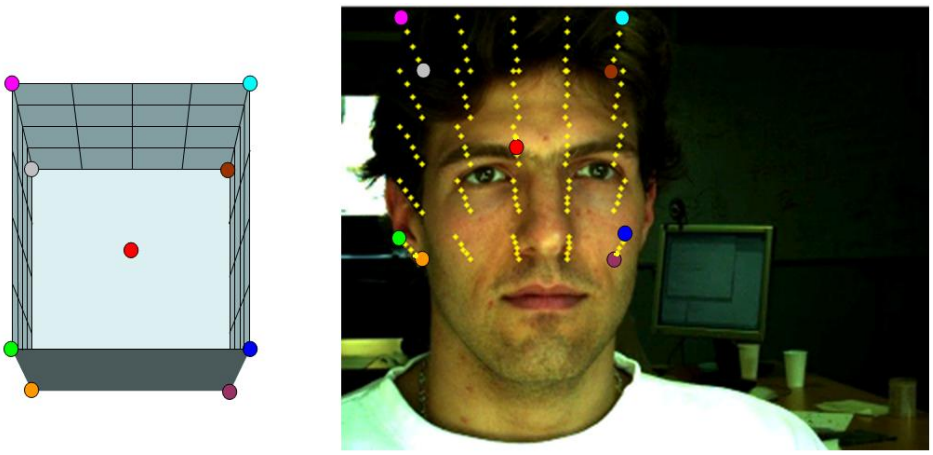


Figure 2.8: Projection of the 3D Grid on the image for one feature. The red dot refers to the current position of the considered control point and to the null translation label. The yellow dots represent the N^3 projections of considered candidates for a control point. This grid is refine in a coarse to fine approach.

interactions between the control points:

$$E(l|\mathcal{S}_{ref}) = \sum_{n=1}^{N} V_n(l_n) + \alpha \sum_{n=1}^{N} \sum_{m=1, m\neq n}^{N} V_{nm}(l_n, l_m) \tag{2.15}$$

l_n and l_m being the labels of x_n and x_m. One should note that this formulation is for a fully connected graph. Such an approach is invariant to translation, rotation and scale. Recovering the optimal solution of this objective function is known to be a NP-hard problem and the complexity is influenced mostly from the pair-wise potentials function. Eventually, the cardinality of the label set is an important parameter since on one hand it defines the accuracy while on the other hand it increases the complexity. In order to address the above mentioned tasks, we consider an approach that is incremental in terms of displacements. Towards computational efficiency and localization of a good minimum, we adopt Fast-PD [65] already introduced in the previous section.

Optimization

The use of a fully connected graph can guarantee the global consistency through local interactions. However, one should note that such a model inherits a lot of redundancy. The distribution of the normalized distance between pair of control points carries on different level of information. Optimizing the pairs of connec-

tions is a challenging problem which can be done using metric from statistics like the mutual information.

In order to further decrease the complexity of the model, we adopt an hierarchical approach in terms of label set, where in each iteration t, and knowing the best configuration at time $t-1$, we are looking for the set of labels that will improve the current solution of:

$$E(l^t|\mathcal{S}_{ref}, l^{t-1}) = \sum_{n=1}^{N} V_n(l_n|l^{t-1}) + \alpha \sum_{n=1}^{N} \sum_{m=1, m\neq n}^{N} V_{nm}(d^{l_n}, d^{l_n}|\mathcal{S}_{ref}^{t-1}) \quad (2.16)$$

with

$$\mathcal{S}_{ref}^{t-1} = \mathcal{S}_{ref} + \sum_{\tau=1}^{t-1} d^{l^\tau}$$

In our minimization scheme, the image support term becomes:

$$E_{im}(l) = \sum_{\mathbf{n}\in\mathcal{G}} \underbrace{f\left(\arg\max_{\mathbf{y}\in\mathcal{V}(\mathbf{x}^{l_\mathbf{n}})} \sum_{t=1}^{T} \alpha_t h_t(\mathbf{y})\right)}_{\approx V_\mathbf{n}(l_\mathbf{n}^{t-1})} \quad (2.17)$$

where $\mathbf{x}^{l_\mathbf{m}}$ is the projection onto the image of m-th control point when translated by the vector d^m associated to the label l_m and f is a positive decreasing function (typically $f(x) = e^{-x}$).

Concerning geometric constraints term for a given point m, it comes to the same thing than minimizing the log-likelihood $-\log(L(D|l))$ and by simplification, finding:

$$\arg\min_{D} \sum_{n=0}^{N} \frac{(d_{m,n} - \mu_{m,n})^2}{2\sigma_{m,n}^2}$$

The geometric constraints becomes :

$$E_{an}(l) = \underbrace{\left(\sum_{\mathbf{m}\in\mathcal{G}} \sum_{\mathbf{n}\in\mathcal{N}, n>m} \frac{(d_{m,n} - \mu_{m,n})^2}{2\sigma_{m,n}^2}\right)}_{\approx V_{\mathbf{mn}}(l_\mathbf{m}, l_\mathbf{n})} \quad (2.18)$$

In [Tab. (2.2)], we present the detection rate for each of the 19 features, for tests on 45 images. We consider that a feature is correctly detected when the distance to the true position is less than 10% of the distance between the eyes. The choice of using a threshold equal to 10% of the inter-ocular-distance was guided by the use this distance in [104]. [Fig. (2.9)] shows some examples of

Feature #	0	1	2	3	4	5	6	7	8	...
Recovering rate	0.96	0.98	0.97	0.94	0.97	0.94	0.92	0.89	0.86	

Feature #	9	10	11	12	13	14	15	16	17	18
Recovering rate	0.97	0.95	0.97	0.94	0.99	0.96	0.91	0.93	0.95	0.83

Table 2.2: Detection rate of our method for each interest point with a tolerance of 10%. A point is considered correctly detected when the distance to the ground truth's position is less than 10% of the distance between the eyes.

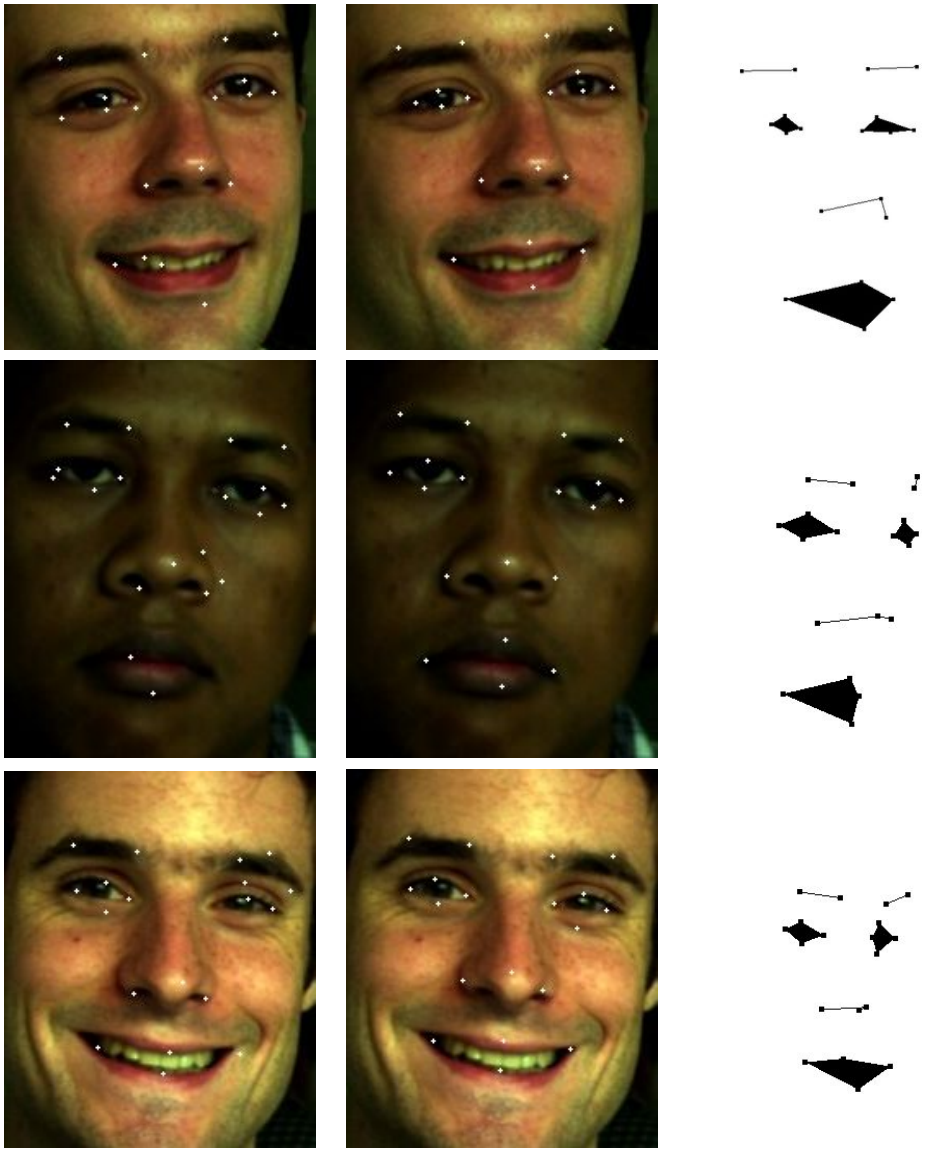
detections with the only Adaboost algorithm (using grey scale and Haar Basis Functions features) and with our method, for different people with different facial expressions. We also provide a 3D representation of the facial mesh. Our method successfully extract the interest points when Adaboost fails to recover a face.

The absolute 3D position is not recoverable, as we use only one static image. Nevertheless, we are able to recover the relative position of the points, which is enough in most studies. In [Fig. (2.10)], we show the image of the error on the normalized distances for each pair. The pixel gray level represents the rate of miss-estimation of the distance between two control points. A black pixel represents a 0% wrong estimation rate while a white pixel implies 100% of false estimation. The diagonal has to be omit, it does not refers to anything: there is no distance between two same points. We consider 10% of the inter-ocular-distance as the error reference. A distance estimation is considered good when the error, compared to the real distance, is smaller than this reference.

This 3D pose estimator was trained, on our 6 basic emotion database with frontal-views only. This is not ideal but the MRF formulation, by taking into account relative positions of interest points, permits to recover the 3D face position even with an unsuited training set. The results were promising according to the 10% of the inter-ocular-distance. But in the context of real world applications these results are not sufficient, as all the points have to be visible, and there are no tilting head (only noding heads). To this end, our facial point detector has to be extended to non-frontal view point detection. It supposes dealing with:

- partially occluded faces and so with invisible points
- tilting head

This implies to have a cascade adaboost classifier robust to rotation, or to be able to determine the tilting of the face (and register it before applying the cascade adaboost classifier), but also a Markov Random Field formulation that doesn't penalize a configuration with one or several invisible points in favor of a 19 points configuration, but with uncorrect detections.



(i) (ii) (iii)

Figure 2.9: Features extraction: (i) using only Adaboost algorithm, (ii) with our method, (iii) the 3D estimation.

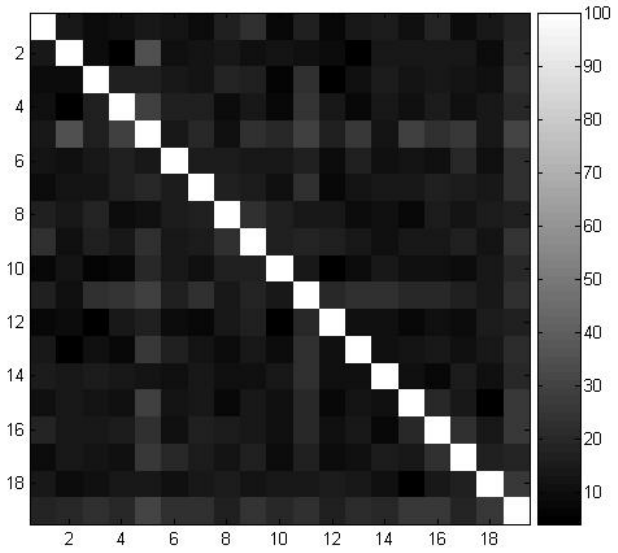


Figure 2.10: Error image of the 3D distances estimation between two control points: in black, good estimation rate, in white, bad estimation rate. The diagonal is meaningless and should be omit: the distance between two same control points is always null.

2.5 Motion Tracking

Motion capture is the way to capture the movement of an element, usually a person, to immerse it and its movement in a virtual universe. A first approach was to place active sensors on the users, such as gloves [3] or sensors attached on the body [77], acting as signals receiver or transmitter. These sensors are in a constant connection with a source or another sensor. This technique requires cumbersome equipment for the user. Passive sensors (like markers attached to the user or make-up [5]), compared to active sensors, don't imply generation of signals or that kind of equipment. Anyway, they require a long and tough preparation. To overcome this, scientists head towards motion capture without any markers, only one or several cameras, processing the images. But the simpler are the sensors, the more difficult is the motion capture. In this direction, in [109] a system for face tracking and animation is presented where facial feature tracking is done in the infra-red space. But even without the use of markers this method require special hardware or material, loosing the advantages of using a sequence from only one camera for the expression mimicking.

2.5.1 Features tracking

Tracking methods, like 3D reconstruction, can be generic for any purpose, or specific to facial features, taking into account the shapes, for example. The most used and known techniques for tracking are optical flow and particle.

As mentioned previously, optical flow refers to the motion field in an image. Given $I(x, y, t)$ the intensity of pixel (x, y) of image I at time t, the point is to define dx and dy, assuming the intensity of a point does not change in time, such as :

$$I(x + dx, y + dy, t + dt) = I(x, y, t)$$

But this formulation makes the optical flow very sensitive to noisy data. On the contrary particle filters, by simulation of possible cases, overcome this problem. Particle filters are receiving a lot of attention for several years now [58, 57, 89] in the tracking of objects in time. Given a set of observations $Y = (y_0, y_1, \ldots, y_t)$, $t \in [1, \ldots, T]$ the aim of particle filtering is to estimate the sequence of state $\alpha_0, \alpha_1, \ldots, \alpha_t$ from $p(\alpha_0)$ and the a posteriori probability $p(\alpha_t|Y)$. The Sampling Importance Resampling algorithm (SIR) permits to estimate this probability by a set of particles $x_{k,t}$ of weights $w_{k,t}$, $k \in [1, N]$. The details are given in algorithm [Algo.3].

In [86], Patras and Pantic first enhance the classic Particle Filter and applied it to the problem of template-based tracking of multiple facial features. They uti-

Algorithm 3 Sampling Importance Resampling algorithm for particle Filtering.

Given a state $\alpha_{k,t-1}$, a set of observations $Y = (y_0, y_1, \ldots, y_t)$.
for $k = 1, \ldots, N$ **do**
  Pick particle $x_{k,t-1}$ with the associated weight $w_{k,t-1}$.
  Propagate $x_{k,t-1}$ to $x_{k,t}$ with the transition probability $p(\alpha_t|\alpha_{t-1})$.
end for
for $k = 1, \ldots, N$ **do**
  Assign a weight $w_{k,t}$ to $x_{k,t}$ such that $w_{k,t} = p(y_t|x_{k,t})$
end for

lize the fact that likelihood can be factorized $p(y|\alpha) = \prod_i (y|\alpha_i)$ in the case that the state α can be partitioned in P groups of random variables (i.e. $\alpha = \{\alpha_i\}$). The proposed scheme can be summarized as follow : First, propagate and evaluate independently the particles partitions, creating a particle based representation $p(x^i_{k,t}|Y)$; Sample from a proposal function $g(x_{k,t}|Y) = \prod_i p(x^i_{k,t}|Y)$; And Compute the particles' new weight, by evaluating the transition probability $p(\alpha_t|\alpha_{t-1})$ so that the set of particles represent the a posteriori probability $p(\alpha|Y)$. Applied to facial features tracking, the authors use a color-based observation model and a priori information of relative features positions regarding to each others. The method is used in [102] for facial action unit detection.

Many other tracking methods, non specific to facial features can also be used for features tracking, particularly those tracking non rigid objects, such as color based methods [88, 111]. Nevertheless, one can have a strong a priori knowledge on the face structure. To this end, model based tracking could constrain the tracking and avoid impossible configurations.

2.5.2 Model Based tracking

Model based tracking methods assume to deform the shape model or face model from a know initial position.

In [76], Malciu and Prêteux use a deformable template-based method, to track the mouth and the eyes contour. A deformable template can be seen as a discrete parametric model describing the object, where the shape is controlled by a set of connected key points, and a set of shape functions to rule the non key points positions. The matching between the template and each frame is established on the minimization of a energy $E = E_{int} + E_{ext}$, where E_{int} encodes a priori constraints on the variability of shape properties (such as elasticity and symmetry), and E_{ext} encodes interaction constraints to maintain the consistency between the geometry and image features (such as edge or texture information).

The features contours can also be modeled by the well known Active Appear-

ance Model [22], already used for face reconstruction or features extraction. In [26], after providing the features position with locally constrained Active Appearance Model, the authors propose to track them. The particularity of the features extraction method, compared to all the other methods based on AAM, is that, instead of reconstructing the whole face appearance, only a patch texture around each feature has to be reconstructed. For the tracking process, the locally constrained Active Appreance Model is reiterated at each frame, initialized with the previous frame parameters.

In a more complex fashion, face model based tracking permits to obtain the rigid and non-rigid global face movement.

DeCarlo and Metaxas address the problem of model-based facial tracking, in several of their works [27, 29, 28]. In their approach, the model, presented in [27], is parametrized by a vector of values q, separated into q_b which describe the basic shape of the object, and into q_m, which describe its rigid and non-rigid motion. The parameters q are estimated based on points u velocities, given by $\dot{\mathbf{x}}(\mathbf{u}) = \mathbf{L}(\mathbf{u};\mathbf{q})\dot{\mathbf{q}}$, where $\mathbf{L}$ is the model Jacobian. The optical flow constraints are :

$$\nabla I_i \begin{bmatrix} u_i \\ v_i \end{bmatrix} + I_{t_i} = 0$$

where, ∇I are the spatial derivatives, I_t is the time derivative of the image intensity and u_i and v_i are the components of the optical flow velocities. They are used to estimate both the rigid and non-rigid motion of the face, such as :

$$\begin{bmatrix} u_i \\ v_i \end{bmatrix} = \dot{\mathbf{x}}_p(\mathbf{u}_i) = \mathbf{L}_{mp}(\mathbf{u}_i)\dot{\mathbf{q}}_m$$

where $\dot{\mathbf{x}}_p(\mathbf{u}_i)$ is the projected model velocities and $\mathbf{L}_{mp}(\mathbf{u}_i)$ is the projected model Jacobian corresponding to the motion parameter.

The Piecewise Bézier Volume Deformation (PBVD) tracker developed by Tao and Huang [98] is used in [20] for face and features tracking. Assuming the features position in the first frame are known, a 3D wireframe model is warped, fitting the landmarks. The tracker uses a model-based approach, consisting of 16 surface patches embedded in Bézier volumes (such that the surface patches are continuous and smooth). The mesh shape can be modified by the displacements of Bézier volume control points. The 3D motions are then estimated from the 2D motions, estimated using template matching between frames at different resolution. In a more technical way, the developed tracker can define the displacements of the face model nodes $\mathbf{V}$ by $\mathbf{V} = \mathbf{BD}$ where $\mathbf{D}$ is the matrix of displacement vectors of the Bézier Volume and $\mathbf{B}$ is the mapping in terms of Berstein polynomials.

In [56], the authors track the whole face in high resolution, and particularly subtle dynamics in expressions, using a face model in a hierarchical framework.

To this end, they need high resolution data, acquired according to [53], and already introduced earlier. The face model is expressed in two resolutions : coarse resolution for global deformations and fine resolution for subtle details tracking. The initialization consists in the manual selection of 30 points on the face, helping the global registration, following a variational non-rigid shape registration algorithm [54]. This algorithm is based on the integration of an implicit shape representation and Free Form Deformations. To track the expression details, the mesh is refined (from 1000 nodes to 16000 nodes). It is registered to the frame, using the same method that was used in the coarse step.

2.5.3 3D Feature Points Tracking

Point tracking comes to the same thing as detecting its position at time t, knowing its position time $t-1$. Once the features position is known in the first image of a sequence, it is easy to track them in the rest of the sequence. Considering our method for 3D pose estimation, tracking is equivalent with deforming the 3D model defined at time $t-1$, in such way that its 2D projection coincides with the features in the image at time t. Considering facial movements are smooth, and the sequence frame rate is high enough to keep this smoothness, it is possible to track the features by considering the 3D grid of features possible displacements, but with smaller values from the very forst iteration. [Fig.(2.11)] presents some examples of features tracking in a sequence.

Feature points tracking in a video, using the points position at the previous frame as initialisation, is a natural extension of feature points extraction in a single image. There is no theorical garanty on the results or on the complexity (linear to the number of frames). But in practice, it improves significantly computational speed. Our raw results are noisy, and one can see flickering. Better results could be obtain, for example, using a Kalman filter.

2.6 Conclusion

We presented in this chapter an overview of different techniques for facial features extraction and tracking.

We propose to combine a texture based approach using Adaboost algorithm on Haar basis function with anthropometric constraints such as the relative positions of features points regarding to each others [46]. To obtain the 3D positions of the features points from a single image and pose estimation, we added prior knowledge for facial features extraction. This problem was formulated as a Markov Random Field problem and solved thanks to the Fast-PD minimization technique.

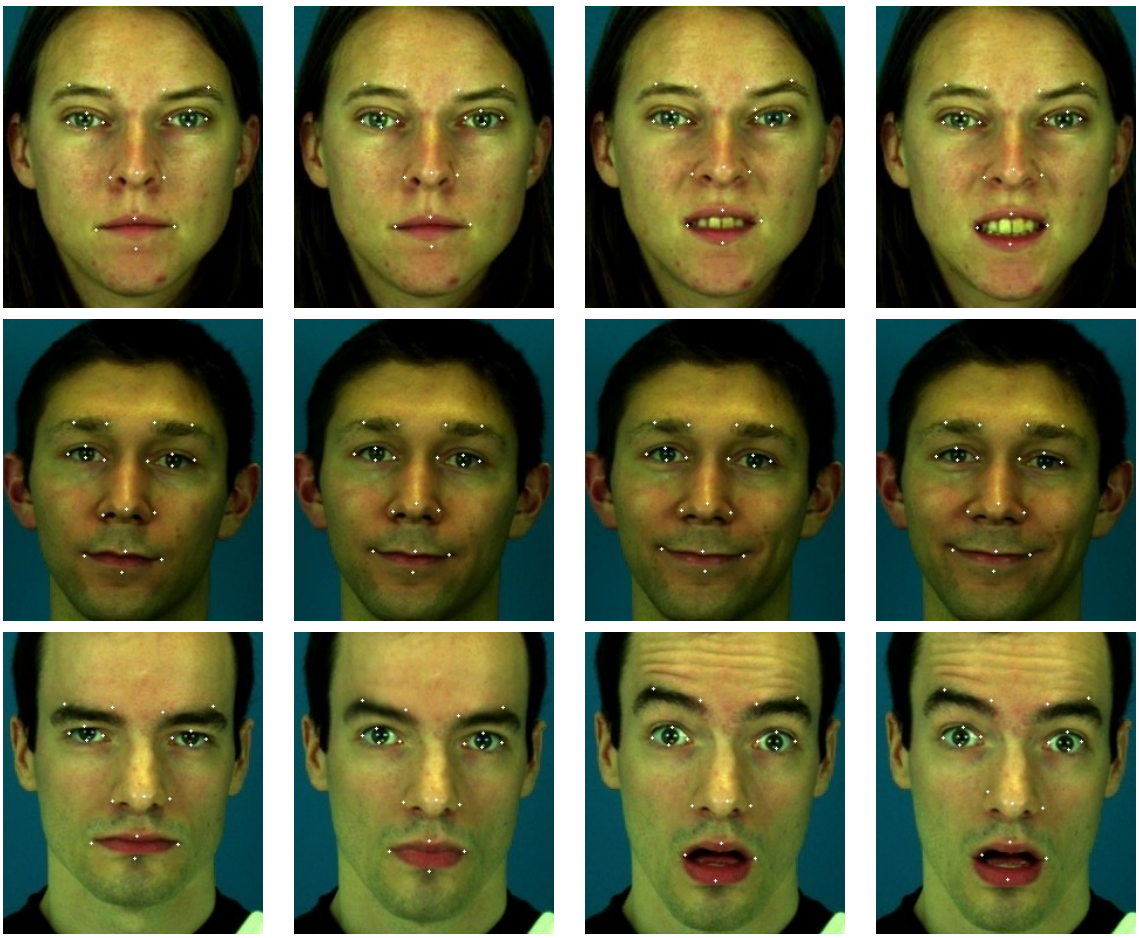


Figure 2.11: Tracked sequence: some frames from a video sequence tracked in a consecutive manner.

This work was extended to facial feature tracking, using the facial feature positions at time t as an initialization at time $t + 1$.

There are several enhancement that could be studied. The pose is estimated in the context of the near frontal and upright view. In real life, the head could be slanting, moving and points insivible. Our method, and particularly anthropometric constraints should be able to deal with these conditions. Then, temporal constraints should be introduced to reduce the flickering in the tracking.

Once the face points of interest are known in an image or in a video, the next aspect to be addressed is their analysis, with the purpose of emotions modeling and recognition.

Chapter 3

Facial Behavior Analysis

In Human Computer Interaction, Expression Analysis is one of the most studied subjects, probably for two major reasons ; First without words, face is the part of the body which reflects feelings the best ; Second because the face is always visible from a webcam put down at the top of the computer screen. Being able to define the characteristics of an emotion and reproduce this emotion, permits to animate an avatar in a video-conference (msn messenger, etc...), in a video games, educative programs on computer (to encourage the student or warn him) or in post-production. But Expression Analysis refers to emotion recognition too. From information extracted in a sequence (appearance, motion,...), the recognition allows to decrypt automatically what a person feels, watching a movie, a commercial, or using a computer, and so, to act in consequence. But behavior analysis doesn't only mean expression or emotion analysis, but also all information the face could give us. For disabled people, the recognition of facial movements could permit to use the face as a mouse. And in a totally different field, the car industry, researchers use the face and more precisely the eyes to detect when drivers fall asleep.

3.1 Emotion Modeling

The term of 'Expression Modeling' refers here to methods to make a face model expressive. The idea is, from a face image or a 3D face model, to make it happy, sad, surprised, ... by expression mapping (also called perfomance driven animation) or using parameters defined by experts or learnt from a database. This is not about 'how to deform a mesh' which was already introduced earlier. In this chapter, we propose to present different methods of Emotion Modeling, for 2D images or 3D models using parameters defined by human or estimated from a corpus or even by performance driven animation.

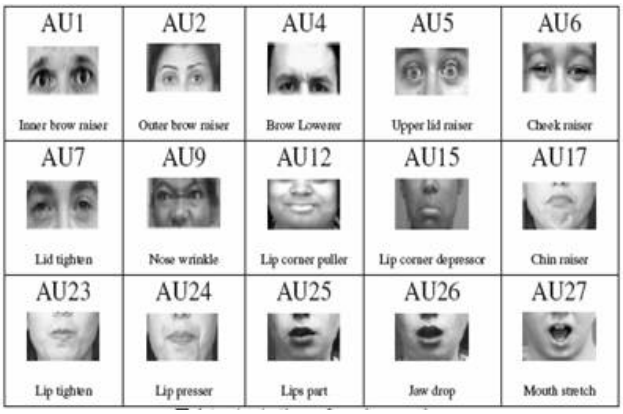


Figure 3.1: Examples of some action units extracted from Cohn and Kanade's database [60]

3.1.1 State of the Art

In the seventies, Paul Ekman and Wallace F. Friesen designed a muscle-based [32] Facial Action Coding System (FACS) to help human observers in describing verbally facial movements using Actions Units (AUs) (or Visemes, comparable to phonemes in speech). Its last version in 2002, includes 46 AUs (inner brow raiser, lower lip depressor, etc...) for facial movements, 12 AUs expressing the global position of the head (Head turn right, head down, etc.), and 4 AUs for the gaze. [Fig. (3.1)] shows few examples of AUs. Each of them is related to the contraction of one or several muscles and any facial expressions can be described as an activation of one or a combination of AUs. As an example, in [83], the authors give a non-exhaustive definition of the 6 basic emotions (anger, disgust, fear, joy, sadness and suprise) through set of the AUs. (See [Tab. (3.3)]

An alternative to that muscle-based understanding of facial expressions through AUs is a description according to geometrical changes of the face. Such an approach is more tractable and reformulates expression understanding in a more technical fashion, and so is suitable for animation. Several geometric models have been proposed such as the MPEG-4 standard.

MPEG-4 aimed at audio and video coding representation, where MPEG-4 [39] is extended to multimedia including images, text, graphics, 3-D scenes and particularly face and body animation. Such an animation mechanism consists of a set of 84 Feature Points (FPs), [Fig. (3.2)] associated with a set of 66 Facial Action Parameters (FAPs), corresponding to a particular facial action deforming a face model (See [Tab. (3.4)] for examples). Facial Action Parameters are based on

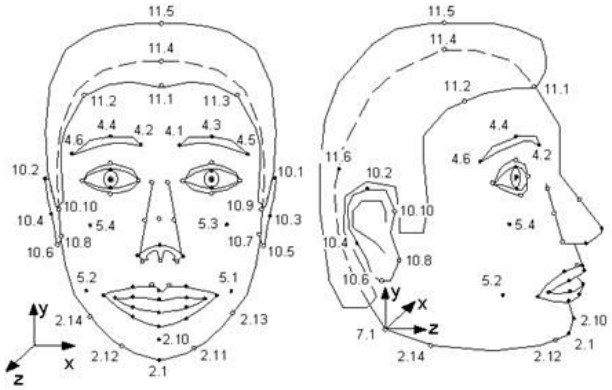


Figure 3.2: The 84 Features Points of the MPEG-4 standard for animation.

the Minimal Perceptible Actions and are related to muscle actions. The 68 Facial Action Parameters are classified in 10 groups corresponding to different elements of the face. One can compare them to the Action Units of Ekman and Friesen's system. But Action Units are used in order to depict a state rather than an action. Furthermore, Facial Action Parameters are more precise in the description. For example, AU #1 correspond to raised inner eyebrows, right and left, while the vertical displacement of inner eyebrows are defined by different FAPs : #31 and #32. Additionally, those two FAPs describe not only how to raise the inner corner of the eyebrows but also how to lower them. Finally, as mentioned previously, MPEG-4 permits to formulate an expression in a technical fashion (See [Tab. (3.5)]). This way, FAPs definition gives also the moving Features Points and the movement amplitude and direction.

Active Appearance Models (AAM) [22] of Cootes et al. are heavily used in Facial Analysis, and particularly in face reconstruction. They have been naturally extended to emotion synthesis. To summarize, a shape s and a texture g can be computed as $s = \bar{s} + Q_s c$ and $g = \bar{g} + Q_g c$ where $\bar{s}$ and $\bar{g}$ are the mean shape and texture, Q_s and Q_g are the matrices depicting the principal variations in shape and texture and c are the specific parameters of s and g for a given face. Kang et al. [61] proposed to define c as a linear combination $c = a_0 + a_1 I$ for expression synthesis. a_0 and a_1 are parameters, learnt by linear regression for each expression. By varying the parameter I they make the expression more or less intensive from $I = 0$ for a neutral face to $I = 1$ for a high magnitude expression. They extend this approach to recompose the neutral face from an expressive face of a known emotion and to expression evolution in a video sequence.

Another way to model an expression and synthesize it on a face, is the de-

Expression	Aus-coded description
Happiness	#6 + #12 + #16 + (#25 or #26)
Sadness	#1 + #4 + (#6 or #7) + #15 + #17 + (#25 or #26)
Anger	#4 + #7 + (((#23 or #24) with or not #17) or (#16 + (#25 or #26)) or (#10 + #16 + (#25 or #26))) with or not #2
Disgust	((#10 with or not #17) or (#9 with or not #17)) + (#25 or #26)
Fear	(#1 + #4) + (#5 + #7) + #20 + (#25 or #26)
Surprise	(#1 + #2) + (#5 without #7) + #26

Figure 3.3: Description of the six basic emotional expressions in terms of AUs according to [83]

composition, as explain by Wang and Ahuja in [106]. Given a database of facial expressions of different people, they also use the well known Active Appearance Model to reduce the dimensionality. They represent the corpus information as a third order tensor A. Using High Order Singular Value, the tensor is decomposed as $A = S \times U^{person} \times U^{expression} \times U^{features}$ where S is the core tensor, U^{person}, $U^{expression}$ and $U^{features}$ are the matrices coding for the person, the expression and the features position. Combination of these subspaces allows to recreate an emotion for a given face, or even, to give another emotion for an already expressive face.

Bettinger and Cootes [9] propose another approach using Active Appearance Model, but for performance driven animation. From a sequence of facial actions, they create a new sequence of actions in a different order. To this end, a sequence of facial animation is tracked using AAM, where the first frame is manually annotated and the initialization of the subsequent frames is made by the previous parameters. This build a sequence of parameters, segmented in terms of actions. A principal component analysis is processed again on this set of actions. The generation of new actions is possible now by sampling the action parameters. Finally, to keep a temporal coherence between the actions, their relations with the source sequence is learned. An extension to 3D could be conceivable according to Blanz and Vetter 3D morphable model [11] or the Combined 2D+3D Active Appearance Models of Xiao et a. [112].

To improve, again, the realism of performance driven animation, Liu et al.

#	Name	Description
3	Open-jaw	Vertical jaw displacement (does not affect mouth opening)
4	Lower-t-midlip	Vertical top middle inner-lip displacement
5	Raise-b-midlip	Vertical bottom middle inner-lip displacement
20	Close-t-r-eyelid	Vertical displacement of top right eyelid
21	Close-b-l-eyelid	Vertical displacement of bottom left eyelid
22	Close-b-r-eyelid	Vertical displacement of bottom right eyelid
23	Yaw-l-eyeball	Horizontal orientation of left eyeball
35	Raise-l-o-eyebrow	Vertical displacement of left outer eyebrow
36	Raise-r-o-eyebrow	Vertical displacement of right outer eyebrow
37	Squeeze-l-eyebrow	Horizontal displacement of left eyebrow

Figure 3.4: Facial Actions Parameters definition provided in [39]

[74] introduce Expression Ratio Image (ERI), to introduce the subtle changes in illumination and appearance due to wrinkles. Usually, the expression mapping is quite simple. From a neutral and an expressive image of the same person, key points on the face are manually or automatically selected. The difference vector is then added to a another face, to generate the same expression [8]. The Expression Ratio Image is computed and used in the following way; According the Lambertian model, the intensity in a point p of the model depends on the m light sources with the intensity I_i, the normal n at point p and the reflectance coefficient ρ such as : $I = \rho \sum_{i=1}^{m} I_i n \cdot l_i$ with l_i is the direction from p to the source light. Then, the expression ratio, can be computed at each point as the ratio between the illumination of the neutral face, and the illumination of the expressive face (as the normal at point p changes and the direction to the light source too, the illumination is not the same). This leads to an entire Expression Ratio Image, which should be approximately the same for any other face, with the same expression. A filtering step is finally added to the ERI to remove the noise, because of the possible shift in the alignment between the neutral source image and the expressive source image.

#	Expression	description
1	Joy	The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.
2	Sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
3	Anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.
4	Fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
5	Disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
6	Surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.

Figure 3.5: Basic facial expressions defined by Facial Actions Parameters in [39]

Thus, it is easy to deduce to new illumination in a synthesized expression image.

Essa and Pentland introduce in [35] and then in [34], a performance driven animation technique based on the optical flow. They derive a facial mesh, developed by Platt and Badler [90], by adding anatomically-based muscles and the elastic nature of facial skin. A mass, stiffness and damping matrices are defined for each triangle, while an attachment point of muscles with the skin is determined. The author use optical flow processing as the basis for perception and measurement of facial motion $\hat{v}(t)$. As the information about motion in depth is not available in a 2D sequence, the velocities in z axis should be estimated. To this end, a spherical mapping $S(u, v)$ is defined, where are u and v are the spherical coordinates. The velocities are so mapped to the face model using a mapping function $M(x, y, z)$. The muscle actuation induced by the mapping is used it for emotion recognition, as I'll present later.

In the context of a 3D model animation, expression modeling refers to motion

of points, face elements, etc. ... Methods like facial Action Coding System or using MPEG-4 animation parameters, involve the displacement of different part of the face separately, without taking into account the face as a whole. One should admit these methods are local, considering that Action Units or Facial Action Parameters are independent, regarding themselves. A global method defining displacements of all features together, would be able to better explain the expression.

3.1.2 Expression Modeling as a Time Serie

We want here to propose dynamic expressions modeling for facial expression, in order to animate our 3D face model, presented in the first chapter. Here, we consider the 6 basic emotions : Anger, Disgust, Fear, Joy, Sadness, Surprise. Obviously, in every day life, these emotions can be expressed differently and actually, there are a lot more expressions. But these 6 basic emotions have to be seen as a starting point of our facial behaviour analysis. [Fig. (3.6)] show them as the deformation of our model. First of all, we focus on a parametric expression model, by modeling the movements of the 19 feature points (the new position of the other mesh vertices is always computed using Radial Basis functions).

We propose a global approach, where the position of each feature points at time t can influence the position of other feature points at time $t+1$.

We model transitions between expressions using an AutoRegressive Process (ARP) [45, 46]. ARP [14] are typically applied to time series data and are used to estimate them. Building these predictive models is equivalent with expressing the state of a random variable $\mathbf{X}(t)$ in time as a function of the previous system using a linear or a non-linear model:

$$\hat{\mathbf{X}}(t) = G(\mathbf{X}(t-k); k \in [1,p]) + \eta(t) \tag{3.1}$$

with p being the order of the model and η a noise model that is used to describe errors in the estimation process. Such a process can be decomposed into a learning stage and a prediction step. In the first step, given a set of sequences of observations and the selection of the prediction mechanism, we aim to recover the parameters of this function such that for the training set, the observations meet the predictions.

The Auto Regressive Process (ARP) permits to solve the problem of predicting objects position in time and is able to model the temporal deformation of a high-dimensional vector. In our context, such a vector corresponds to the position of the face control points (we assume no changes in depth/orientation of the face). A system to model the transitions between expressions could be express such as :

$$\hat{\mathbf{X}}(t) = \sum_{i=1}^{I} W_i f(\sum_{k=1}^{p} w_{i,k}\mathbf{X}(t-k) + \theta_i) \tag{3.2}$$

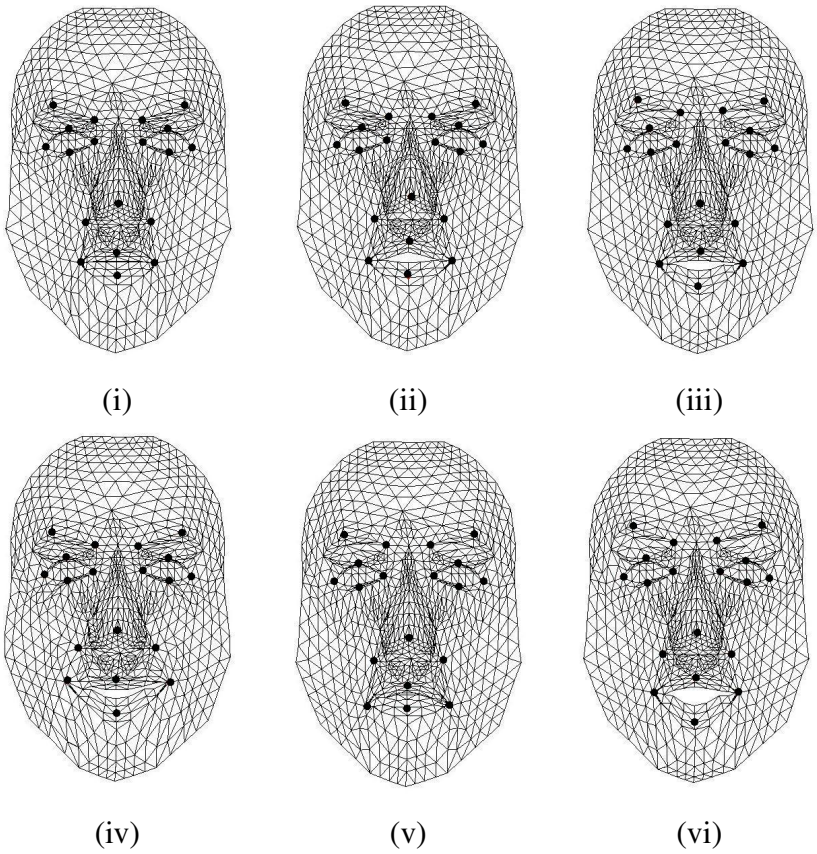


Figure 3.6: The 6 basic emotions : (i) anger, (ii) disgust, (iii) fear, (iv) joy, (v) sadness, (vi) surprise.

where W_i and $w_{i,k}$, $i \in [1, I]$ and $k \in [1, p]$ are the auto regressive process parameters, while $f(x)$ is the identity in case of a linear model or a smooth bounded monotonic function for a non linear model.

Let us now consider, $\mathbf{P}(t)$ the N 3D points position vector at time t, composed as follow :

$$\mathbf{P}(t) = [P_{1,x}(t)P_{1,y}(t)P_{1,z}(t) \dots P_{N,x}(t)P_{N,y}(t)P_{N,z}(t)] \tag{3.3}$$

where $P_{n,x}$, $P_{n,y}$ and $P_{n,z}$ are the coordinates of point P_n. Toward introducing explicit constraints driven from the facial geometry, we compute the correlation coefficients between points in all emotions. Such a coefficient between two points is high when a relation between the behavior of those two feature points can be observed among all emotions, like for the corners of the mouth of the eyebrows.

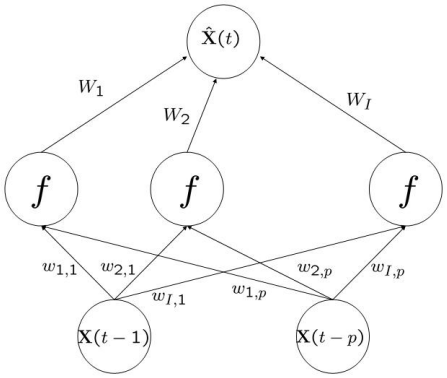


Figure 3.7: Neural network modeling the non-linear auto regressive process. The neural network is fed by the 3D coordinates of feature points from time $t-1$ to $t-p$ to estimate $\hat{\mathbf{X}}(t)$ the 3D coordinates of feature points at time t.

If the coefficient is high enough, the position of the points are assumed to be correlated and mutually dependent during the calculation process. Any element of $\mathbf{P}(t)$ can be so estimated such that:

$$P(t) = \sum_{i=1}^{I} W_i f(\sum_{k=1}^{p} \sum_{m=1}^{M} w_{i,k,m} P_m^*(t-k) + \theta_i), \tag{3.4}$$

where $\mathbf{P}^*$ is the set of M elements of $\mathbf{P}$ such that the correlation between P and any element P_m^* of P^* is above the threshold.

In the case $f(x)$ is the identity, the system is linear and we use a least mean square method to estimate the autoregressive process parameters. If the system is non-linear, it can be modeled by a neural network [Fig. (3.7)], where the input layer is composed of the variable $P^*(t-k)$ with $k \in [1,p]$ and output layer is $P(t)$.

We use a back propagation technique to estimate the parameters. During the initialization, a random weight is assign to each neuron. The training consists of setting the input layer with all the training data successively. At each step, the value of the output layer is computed and compared to the real value. The weights are adjusted to enable the neural network to reproduce the expected output. This process is iterated several times to obtain the better suited network for each emotion.

In [Fig. (3.8)], we present the evolution of displacements of some typical features. More precisely, we show that the difference of position in y-direction between time $t-1$ and time t for the feature 5, when expressing anger, feature 15 when expressing joy and feature 18 when expressing surprise (One should note

Algorithm 4 Back-Propagation for Neural Network Parameters Estimation

```
Initialize the network with random weights
repeat
  for example e in the training set do
    Compute the output by propagating the information forward
    Compute the error
    Adjust the weight backward from output layer to input layer
  end for
until Stopping criterion is not satisfied
```

that the origin of the system of coordinates is at the top left corner of the image). Those graphs prove that an emotion cannot be seen as a linear process, and that the $f(x)$ of the Auto Regressive Process should not be linear.

3.1.3 3D Database of Facial expression

As far as we know, no 3D Expression Database, giving the 3D positions of feature points in time, is available. However, such a database is required to model facial expressions in 3D : we created one. The set-up [Fig. (3.9)] is composed of two cameras, one above the other. In this manner, the face is captured with a frontal view and a view from underneath. This configuration permits to avoid face occlusion by the nose and to keep the symmetry. [Fig. (3.10)] shows examples of stereo images of the database. The images are 480×640 images and sequence is captured at 15 fps .

The database consists of 13 subjects performing the 6 basic emotions. To obtain the 3D positions of feature points in time, first the 19 points are manually selected in the frontal view of the face. Their positions in the underneath view is deduced from the disparity map, computed with the Graph Cut algorithm [16, 64, 15]. The tracking is made by optical flow. As the facial movements are more deformations than movements, we choose to use a block matching optical flow.

Block matching Optical Flow considers that the image is divided in small, overlapping or not, blocks of a given size. The point is to find the most similar block in the following image. In this way, the pixels in the same block move with the same offset. Once the features positions are known in both images, their 3D positions are recovered using the projection matrix.

The result of this preprocessing is, for each person and for each emotion, a set of 3D point coordinates at different time, starting from a neutral face to an emotion.

One should note that 13 people for such a database is not enough to proceed to a complete evaluation of a method/technique. These 13 persons are not actors,

and the emotions were totally fake. Several other points could have been improved also, such as the illumination, the frame, etc... Last but not least, this database was only available at the end of the thesis, and we had no choice than using it just as it was.

3.1.4 Expressions Modeling on our Face Model

We present here each of the six basic emotions at different times, first on our generic model [Fig. (3.11)] and [Fig. (3.12)]. One should note that the texture is computed at the beginning, during the registration of the model on the 3D surface. This texture, defined for each triangle of the mesh, is displayed in the entire sequence.

3.2 Emotion Recognition

Facial Behavior Recognition techniques concern in this section, techniques being able to make the difference between expressions, to distinguish the six basic emotions (anger, disgust, fear, joy, sadness and surprise), or even better, to detect Action Units as the ones defined in the Facial Action Coding System [32]. Techniques could be different, non only according to the classification method, but also according to the extraction of information to be classified. Here, we make a non-exhaustive presentation of techniques, as different as possible. This is not a survey based only on recognition rates and the purposes of recogntion (Emotion or Action Unit ?, static or dynamic ?, how many ?, which database ?) are most of times different. It could be a little bit considered as out of field but one should note that multimodal methods are emerging also those last years, taking into account the body and the face [48], the voice [119], ...

There are many ways to put Emotion Recognition in categories. One can present them according to different aspects :

- Information Sources : The emotion recognition can be made from a static image or a sequence.
- Extracted Information : The information extracted from the sources can be a feature-based approach as well as following a holistic approach.
- Classification technique : Many different approaches are available for classification, such as Hidden Markov Model, Adaboost or rules based classification.

We present here different techniques of emotion recognition, according to the classification technique.

We have already seen the Facial Action Coding System, first developed in 1976 by Ekman and Friesen [32]. The system is based on static position of face components, and the emotions, based on it, are defined in the same way. But in 1979, Bassili [6] showed that facial motions are more important than only facial components positions for facial analysis and thus emotion recognition. That's why here, we focus on dynamic emotion recognition (from a sequence).

3.2.1 Hidden Markov Model

The Hidden Markov Model is a statistic model, considered as a Markov process with unknown parameters (See [Fig. (3.13)]). They are heavily used as temporal process such as in speech recognition, and for several years in emotion recognition. The purpose is to determine the hidden parameters or state S_t from the observable parameters O_t at time t, knowing the transition probability between S_t and S_{t+1}, observation probability distribution and the initial state distribution. Building the HMM comes to defining the transition probability, the observation probability distribution and the initial state distribution.

As a natural approach to track an object in time, optical flow is also used to model expressions. In [80], Otsuka and Ohya propose to describe an emotion in two Image Processing steps. First, they compute a velocity vector between subsequent frames using optical flow in squared regions of the right eye and the mouth. The second step consists in a two-dimensional Fourier transform of the previously computed velocity vector. The lowest frequency components of the Fourier coefficients are then used as emotion descriptors. Then, they feed a classical Hidden Markov Model framework (one HMM for each emotion) to classify the emotion.

Following the idea of features combination, the Bézier Volume presented in [20] and already used for face and features tracking (see section 2.5.2) can define the displacements of the face model nodes $\mathbf{V}$ by $\mathbf{V} = \mathbf{BD}$ where $\mathbf{D}$ is the matrix of displacement vectors of the Bézier Volume and $\mathbf{B}$ is the mapping in terms of Berstein polynomials. It permits to model each facial expression as a linear combination of facial movements : $\mathbf{V} = \mathbf{BDP}$ where $\mathbf{D}$ is the concatenation of $\mathbf{D_i}$ matrices, corresponding to a facial movement and P is the set of magnitude of each deformation. The authors tested two different uses of HMM. The first one is the most natural one : modeling each expression with a HMM trained on that expression. The observation considered O_t is the continuous face deformation D and the hidden states are, obviously, the expressions. This method supposes there is only one expression in the sequence, or there is a pre-segmentation of the sequence, like in most recognition techniques. To overcome this, they propose to use multi-level HMM for the automatic segmentation of the sequence and the recognition. The framework can be describe as follow : the motion data (represented by the matrix D in the Piecewise Bézier Volume Deformation presented

earlier) are continuously used as input of the 6 emotion-specific HMMs, as described previously. The output is then used as the observation vector for a high level Markov Model, consisting in 7 states (6 emotions and the neutral state).

Lien et al. in [72] propose to extract 3 different informations from an image sequence to recognize the upper face Action Units, using one Hidden Markov Model for each of these Action Units. The first source of information is the movement of facial feature points (6 points), manually selected in the first image and tracked by optical flow. The displacement in each frame are registered by subtracting the normalized positions in the first frame by the actual normalized positions. The resulting extracted information is a 12-dimensional vector corresponding to the displacement in vertical and horizontal direction of the 6 points. But, this way, the information outside the selected area is totally forgotten. To tackle that, the authors propose as a second source of information to use the dense flow of the forehead. But, due to the high dimension of the data (one velocity vector for each pixels of the forehead, in every frame), the information is compressed using Principal Component Analysis. The authors use a third information source : the high gradient component. Actually, they refer to the transient wrinkles appearing on the forehead when raising the eyebrows, for example. This generate a binary mask of the forehead, which is then divided in 13 regions. The third information used for the recognition is the 13 mean values and 13 variance values coding the 13 binary masks. This techniques could be considered as hybrid, since it uses feature based informationss and template based information.

3.2.2 Support Vector Machine

Support Vector Machine (SVM) is a discrimination technique and consists in separating in two or more, a set of points by a hyperplan (See [Fig. (3.13)]). SVM is based on the use of kernel, permitting an optimal separation.

[102], [70] use Support Vector Machine, with different information extraction, to respectively detect Facial Action Unit and classify emotions. While in [101] is presented a combination of SVM and Hidden Markov Model.

Liao and Cohen propose, in [70] and [71], to compactly define the emotion by a region-based model. The face is divided in 9 non-overlapped regions, in which the movement is assumed to be homogeneous. This assumption permits to describe the motion in each region by affine parameters. Based on this, the facial expression can be defined as the combination of 9 sets of affine motion parameters. To encode the interaction between the regions, the authors propose a graphical face model, with a graph $G = (V, E)$ where the vertices V are the 9 regions and the edges E encode the interaction between the region, enforcing the spatial structure and the symmetry of human faces. In [70], the authors propose to classify expression, combining their graphical model and a SVM process. To

this end, the information extracted from the sequence have to be represented in a hyperplan. The classification problem can be formulated as : $P(x|X) \propto P(X|s) \cdot p(s)$ where s is the variable indicating the class of expressions and X is the vector of extracted motion parameters. It comes to the same thing than compute the log-likelihood function of X : $L(X)$. Since the face has $9 \times 6 = 54$ parameters, to simplify $L(X)$ estimation, the authors propose to split the 6 affine parameters into 3 sets. They obtain 3 log-likelihood values (L_1, L_2, L_3), viewed as point coordinates. They finally use a linear-kernel Support Vector Machine to classified the emotion.

In [102], Valstar et al. present a system for the detection of 16 Action Units of the Facial Action Coding System using Support Vector Machine. Their technique differs from the previous one, by the information to be classified. They propose to detect activation of AUs by motion patterns of 20 fiducial facial points. The facial features, detected or manually selected in the first frame, are tracked from the first image using particle Filtering with Factorized Likelihoods [86]. The information extracted from the tracking is based of the registered feature positions in the sequence. According to the rules of Action Units activation, the data to be classified correspond to euclidean distances between two points in the first frame, and in the subsequent frames (relatively to the first one), and vertical and horizontal displacement of point (always relatively to the first one). Let's note that the author implemented the Probabilistic Active learning Algorithm to train the SVM classification and deal with the large amount of data.

In [95], the author propose to use different kernel for Support Vector Machine. The authors extract information from images using an Active Appearance Model features based on 58 landmarks, and proceed a SVM classification using linear, polynomial and RBF (Radial Basis Function) kernels. One SVM is created for each expression to separate the AAM parameters and the final classification is processed by giving the features to each classifier successively. The kernel used in classifiers for each expression depends on the result in the testing phase. The authors propose to go further, using the same technique to detect the gender of the subject, and even to detect first the gender, then the emotion and inversely. The best results are obtained by a gender classification followed by an emotion classification. This could give another orientation to expression recognition, assuming that emotion are not expressed similarly for men and women.

3.2.3 Neural Network

The neural network (or artificial neural network) is a mathematical model, inspired by how the human neurones work. They were already introduced in the previous section.

In 1996, Rosenblum et al. [93] present a radial basis function network ar-

chitecture to learn the correlation of facial feature motion patterns and human expressions. They pre-process an input sequence as follows ; First, face features are tracked using optical flow, assuming that the regions of interest are given at the first frame. They develop a region tracker for rectangles enclosing the face features. The tracking algorithm integrates spatial (based on the gradient magnitude of the intensity image) and temporal information (based on the optical flow field) at each frame. The spatial tracking of the face regions is based on computing two sets of parameters at the points with high gradient values within the rectangle that encloses each feature. Then the optical flow fields are separated in 4 (one for each principal motion direction : up, down, left and right), and converted in a polar coordinates system. A hierarchical approach is used to identify expression. The first level is a collection of neural networks, one for each expression. Then in each of them, the facial motions are decomposed according to the facial components. Finally, the last level refers to the motion decomposition in the 4 directions. A final interpretation process performs a fusion of the information and give the emotion.

In [123], Zhang presents experimentations for Facial Expression Recognition using a multi-layer perceptron. Two kind of information are extracted from a face image. First 34 fiducial points are selected to represent the facial geometry. Then, a set of multi-scale and multi-orientation Gabor Wavelet coefficients are extracted at each of the previous ficudial points through image convolution. These two sets feed the perceptron, while the first layer performs a non-linear reduction of the dimensionality of feature space. The second layer makes the classification between the 7 output units, giving to each of them an estimation of the probability to belong to the corresponding expression. The author notes that the number of hidden units which fits better the emotion classification is from 5 to 7. But the most interesting information is the relevance of the geometric position, compared to responses to Gabor Filter. Particularly, above 5 hidden units in the perceptron, the geometric positions don't give more information about the expression.

In [99], Tian et al. proposed to use non only the usual features (eyes, mouth, eyebrows, ...) for Action Units recognition but also the apparition of wrinkles or furrows, called here transient features. Indeed, they point that transient features appear perpendicular to the motion direction of the activated muscle and provide crucial information for the recognition. Several states are defined to depict the permanent features (for example, the mouth can be opened, closed or tighly closed) while the transient features can be present or absent. Those states are modeled as geometric pattern to model feature's location, shape, and appearance. At the first frame, the face is assumed to be in neutral state, and the templates are manually selected. They are tracked in the subsequent frames. For the Facial Features Representation, the upper face features and lower face features are represented into 2 groups of suitable parameters, defined by parameters coding shapes, motions and the states of the components. The authors proposed to use neural networks

to classify the upper face AUs and lower face AUs, considering AUs alone or combination of AUs.

3.2.4 Adaboost

Adaboost classifier has already been presented for face 1.2.3 and facial features detection 2.2.1. As a reminder, we can briefly summarize Adaboost classification as the result of the weighted combination of weak classifiers:

$$sign(H(\mathbf{x})) = \sum_{i=1}^{n} \alpha_i h_i(mathbfx)$$

where the sign of $H(mathbfx)$ gives the classification of $mathbfx$, a vector of data to classify, $h_i()$ is one of the n classifier and α_i is the associated weight. Whitehill et al. [110] uses Adaboost to determine the Action Units expressed in an image. First, the eyes and the mouth are manually selected, and their position is used for the image normalization such that their positions and the scale is the same in all images. Then, the dimensionality is reduced, by segmenting the face in regions (mouth, eyes, eyebrows). The classification is performed on Haar filter responses. The Adaboost classifier used is a binary classifier, such as one classifier has to be determined for each Action Unit.

Most of times, Adaboost is used in detection or recognition in a static image. Yang et al. [115] extract and process information from a sequence to obtain vectors suitable for an Adaboost classification. The use of Adaboost in a temporal classification scheme is something unusual and the way to represent the dynamic aspect of the emotion is really important. They exploit the Haar-like features to represent face image, and extend them to represent the dynamic characteristic of facial expressions and Action Units. The information extraction can be summarized this way : in all the images of a sequence, the same set of N Haar-like features are computed. They propose a binary coding system based on statistical distribution of the training samples, to decrease the data dimension. In a training phase, for each AU/emotion, the gaussian distributions $\mathbb{N}(\mu_j, \sigma_j)$ of Haar-like features j are computed. This way, for an image i, the haar-like feature can be coded by 0 or 1 using the following formula :

$$C_{i,j} = \begin{cases} 0 & : \text{ if} \frac{||h_{i,j} - \mu_j||}{\sigma_j} > T \\ 1 & : \text{ if} \frac{||h_{i,j} - \mu_j||}{\sigma_j} < T \end{cases}$$

where T is a threshold. The same feature in all image of a sequence can be easily coded by a succession of 1 and 0, itself coding for a single value in base 10. The Adaboost algorithm use this new values (one for each haar-like feature) for the training and to classify the data.

3.2.5 Belief Propagation

Belief propagation (BP) is a fast optimization method for inference problem, based on passing local messages and designed for connected graphs. BP are commonly used in pairwise Markov Random Fields. In this framework, the idea is to pass a message $m_{i,j}$ during several iterations, from a node i to one of its neighbors j, coding the influence of i on j. At each step, the message is updated. At the end of the process, the algorithm returns the most probable state for each node, according to the observation.

In [71], the authors proposed a pairwise potential $\psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j)$ where $\mathbf{x}_i$ are the affine motion parameters, or, in belief propagation term : the state to be estimated. They add a data term $\phi_i(z_i, x_i)$ measuring the error between the optical flow $\mathbf{z}_i$ observed in the region i and $\mathbf{x}_i$. The message updating equation is :

$$m_{ij}^t(x_j) \propto \int \psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \times \phi_i(z_i, x_i) \times \prod_{k \in N(i)j} m_{ki}^{t-1}(x_i) dx_i$$

where t denotes the iteration, and $N(i)$ is the set of i neighbors. The final marginal distribution $P(x_i|Z)$ at iteration T can be computed as follows :

$$P(x_i|Z) \propto \phi_i(z_i, x_i) \times \prod_{k \in N(i)} m_{ki}^T(x_i)$$

By formulating the classification process by the Maximum A Posteriori formulation, the Belief Propagation framework permits to solve :

$$\hat{c} = \arg\max_c P(c|Z)$$

where c defines the emotion variable. Last but not least, the author investigate the case of occlusion by proposing modifications of the Belief Propagation framework, for the incomplete observation case.

3.2.6 Rule Based

Another approach for Emotion Recognition, is to use rules about feature points motion to determine the expression.

Yacoob and Davis [114] first, then Black and Yacoob [10] propose this approach [10]. Yacoob and Davis use the method already presented in [93] to extract the information from a sequence using optical flow. One should note that, in this case, the information is not translated in polar coordinates. Using a rule based system, instead of a connexionist approach, they define the expression. Black and Yacoob enhance the previous method, first by dealing with important global head

motion. The image motions of the facial features are modeled relatively to the head motion using different parametric models. These motions are modeled using image flow models and are estimated over an image sequence using a robust regression scheme. Thus, they first derive the motion parameters to describe the observed facial changes at each frame (Mid-Level Representations). Then, following the rules of each expression, they are able to divide each facial emotion into two temporal segments: the beginning and the ending (High-Level Representations).

This is the case in [82], where Action Units of Facial Action Coding System, and not only emotions, are determined, using the displacement in terms of direction, of feature points in a profile view sequence. The initial position of the features can be defined manually or automatically, and are tracked using an adapted particle Filtering algorithm, Auxiliary Particle Filtering [89], using a color-based observation model invariant to global illumination changes. The temporal rules permit to encode 27 AUs, alone or associated, and to detect the beginning, the peak or the ending of the AUs. Once AUs are detected, one can use other rules, also available in Facial Action Coding System to determine the emotion.

3.2.7 Distance to a model

In [33], Essa and Pentland propose two methods, using two different information sources for emotion recognition. They first use the information given by the performance driven animation in [34]. The emotion are divided in 3 phases (application, release and relaxation) and normalized such that the expressions occur with the same duration. The similarity of the observed application and release phases with the learned ones is computed by a dot product. For the second approach, the author deduced, from the previous analysis, physical information like the muscles actuations. They use motion energy templates to encode the expressions. Only motion and velocity measurements are needed be to computed. For each expression, the ideal 2D motion energy are used to characterize the spatio-temporal templates. To recognize the expression, the authors used the Euclidean norm of the difference between the motion energy template and the observed image motion energy.

Donato et al. in [30] compare several techniques, based on data analysis, to represent facial expression with the purpose of classifying facial actions. Particularly they analyze holistic spatial analysis vs. local spatial analysis. The holistic analysis begins by the extraction of information in the upper and lower-face sub images at each frame of the sequence. The authors propose to subtract the first image (where the face is assumed to be neutral) to all the subsequent images and to use a matrix of image differences to classify the data. They test 4 types of data analysis methods to represent the information in the image differences :

- Principal Component Analysis (PCA) : The $p = 30$ principal components are obtained from the decomposition of the covariance matrix S into $S = PDP^T$ of the training data, in eigenvectors. During the classification step, the matrix of image difference is mapped into the first p eigenvectors of P, producing p coefficients for each image.
- Local Feature Analysis (LFA) : LFA [87] is a local method since it constructs kernels detecting local structures, and a derivative version of PCA. It reposes on a set of kernels, $K = PVP^T$ where $V = D^{-\frac{1}{2}}$. The kernels were found to have spatially local properties and are indexed by spatial location. The kernels number is specified by a sparsification algorithm to decrease the dimensionality of the representation.
- Fisher's Linear Discriminant (FLD): FLD [38] finds the linear combination of features which best separates objects in two or more classes. It maximizes the distance between the means of the two classes while minimizing the variance within each class.
- Independent Component Analysis (ICA) : ICA is another variant of Principal Component Analysis and is developed for solving the problem of Blind Source Separation(BSS). The set of image differences X is assumed to be a linear combination of unknown image sources S weighted by the unknown mixing matrix $A : X = AS$ The sources S are recovered by the learning of the unmixing matrix W, approximating A^{-1}. One should note that, despite the holistic approach, the basis images are local.

The comparison of these methods are made, for each of them, using the best similarity measure (Euclidean distance or cosine similarity measure) and classifier (template matching classifier or nearest neighbor classifier). Independent Component Analysis clearly outperforms the other techniques.

According to Padgett and Cottrell's work [81] on face representation for emotion recognition, principal components of image subregions containing mouth and eyes are more effective than global PCA as well as shift-invariant local basis functions on small image patches. Donato et al. take this into account for the global analysis. Like in the holistic case, they use image differences :

- Local Principal Component Analysis - Random Location: Several thousands of patches are taken randomly from the image differences, and decomposed using PCA. The first p component are used as convolution kernels to filters the full images.
- Local Principal Component Analysis - Fixed Location: The location of m patches is fixed, and the p principal components of each region are computed, and contrary to the previous local PCA, the principal components are only used at the fixed patches location.
- Gabor Wavelet Representation : Gabor filter is a linear filter defined by the product of a Gaussian filter and an oriented sinusoid. They use the outputs

of the convolution between the image of differences and Gabor kernels, using different orientations and frequencies (working at different frequencies comes to the same thing that changing the resolution of the image). Each of them is down sampled to reduce the dimensionality and normalized to unit length.

– Principal Components Analysis jets : To obtain the same multi scale approach than with the Gabor wavelet representation, the authors propose a multi scale version of the local PCA representation (Random Location version), called PCA jets.

Among those 4 representations, the Gabor Wavelet one give the best results, comparable to the Independent Component Analysis. Even if the ICA is a holistic approach, we should remember that basis images are local. Thus, we can conclude that local spatial filters are important. Nevertheless, the classification used are really simple. A comparison of more complex classifiers on Gabor Wavelet and ICA representation could maybe improve the results.

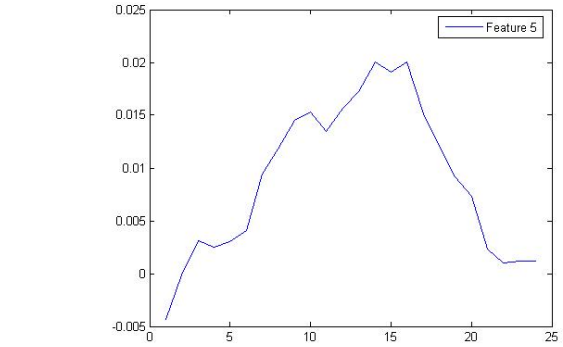
All these methods follow the same scheme : information extraction and classification. While it is possible to model the expression in time, the system is also capable to predict the information (such as features positions, motion field). Making a comparison between the prediction and the real features positions can help recognizing the expression.

3.3 Conclusion

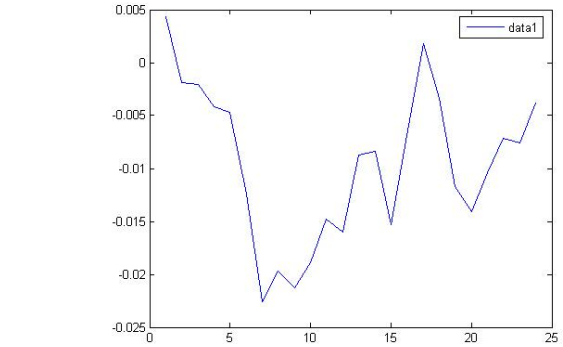
We presented the state of the art and our work in expression modeling. We propose to model the 6 basic emotions with an Auto Regressive Process [45] through a neural network [21], feed by the feature point positions at previous time. One should note that this is only a first step in emotion modeling. To be able to reproduce any emotion or facial movement, the FACS Actions Units should also be desinged through an Auto Regressive Process.

According to this, emotion recognition could be consider following a prediction based method. The point is to predict the positions of feature points in the subsequent images using the Auto Regressive Process and to compare these positions with the real position, detected through the anthropometric constraints and prior knowledge based tracking presented in the previous chapter.

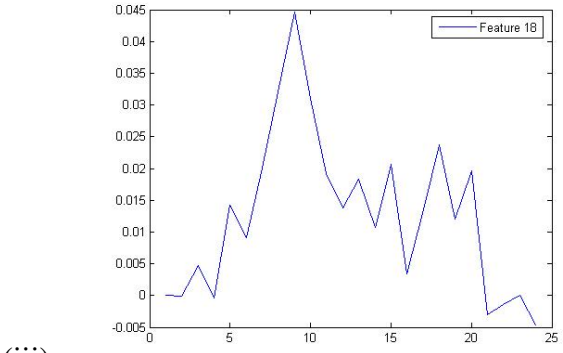
One should be aware that a lot of problems are not addressed in most of the emotion recognition techniques : moving face, partially occluded face, temporal segmentation, emotion beginning by the neutral state, etc .... Particularly, no method deals with all of them. The emotion recognition should then face to the same problems than facial features extraction and tracking like moving head and partially occluded face.



(i)



(ii)



(iii)

Figure 3.8: Graphs presenting the non-linearity of features displacements during an expression : (i) Vertical displacement of an inner eyebrow when expressing anger, (ii) Vertical displacement of a mouth's corner when expressing joy and (iii) Vertical displacement of lower lip's mid-point when expressing surprise. (One should note that the origin of the system of coordinates is at the top left corner of the image)

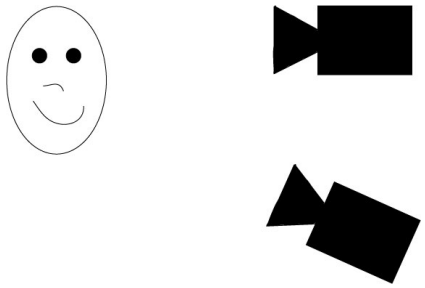


Figure 3.9: Set-up for emotional sequence acquisition in stereo with a frontal view and an underneath view.

Figure 3.10: Examples of stereo images of the database.

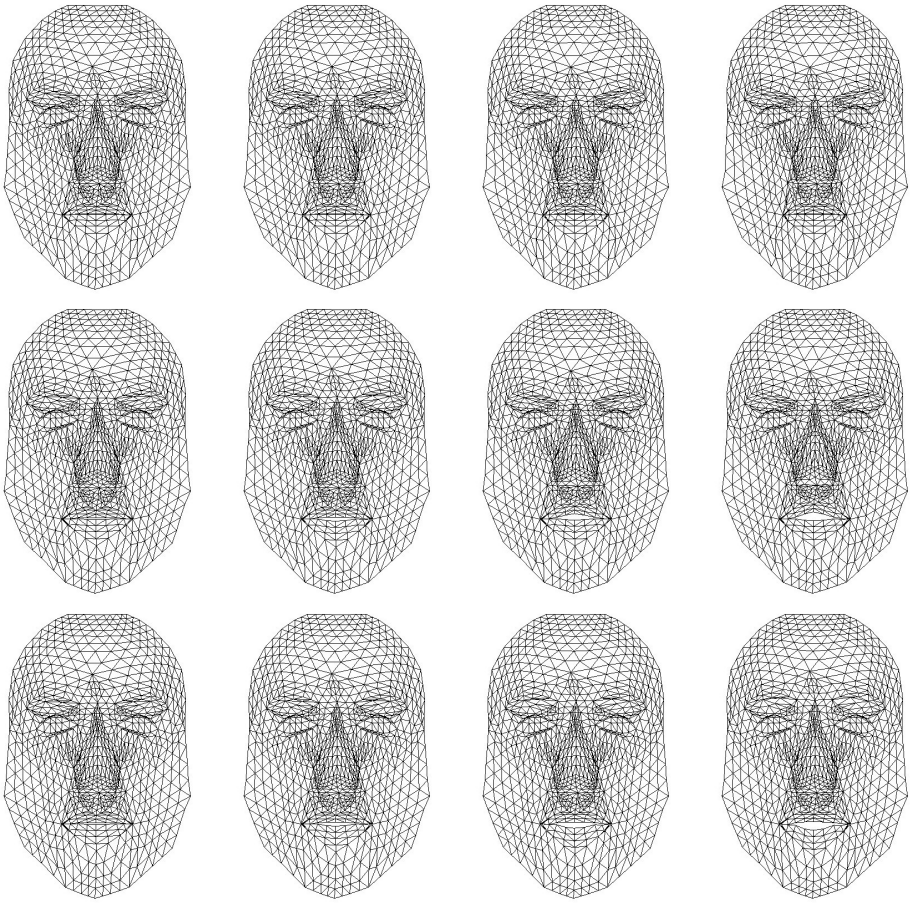


Figure 3.11: Anger, disgust and fear on our generic model, at different times.

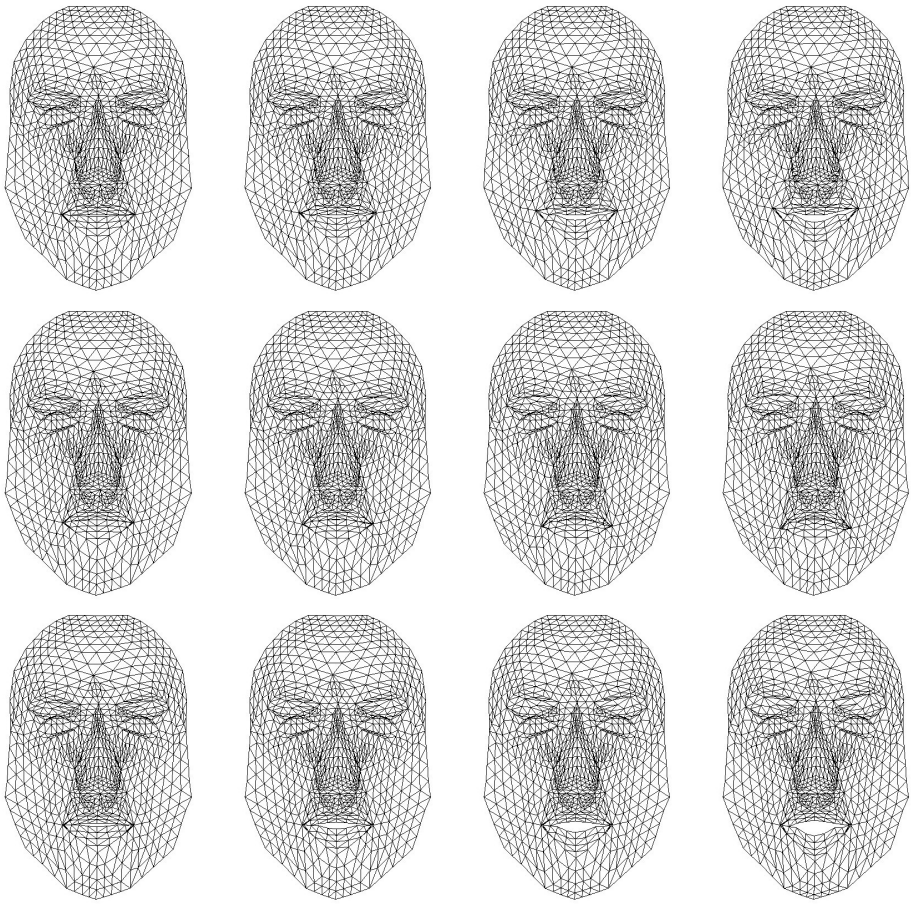


Figure 3.12: Joy, sadness, surprise on our generic model, at different times.

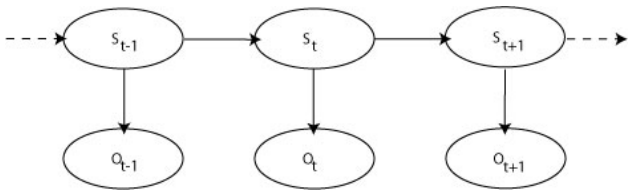


Figure 3.13: Modelisation of the Hidden Markov Model where S_t are the hidden parameter at time t and O_t the observation at time t.

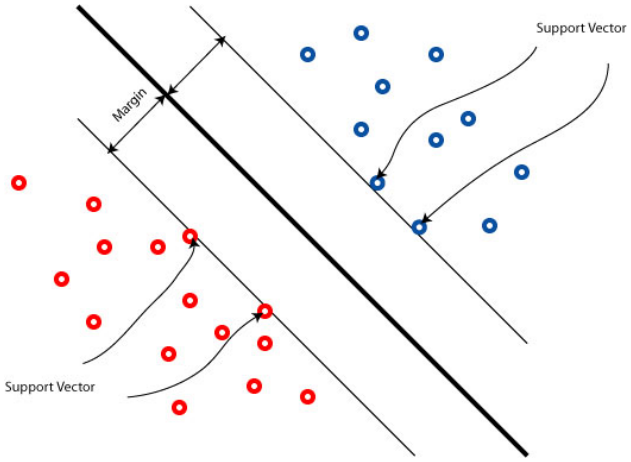


Figure 3.14: Modelisation of the Support Vector Machine

Conclusion

In this thesis, we addressed most of the fields related to the face in Computer Vision. First we proposed a facial model in the form of a mesh, derived from the Candide model as the refined mean of Candide registration to 3D face database [46]. It is sharp enough to be realistic but simple enough to adapt it easily to a given person. We added animation parameters to the model, which consist in a set of control points, associated to a set of points they influenced. The motion of the mesh vertices is defined through radial basis functions and control points motion. In parallel, we propose a technique to recover the 3D geometry of a face that we called super resolution reconstruction [44]. The point of this reconstruction is to discretize the disparity range to a half pixel and solve the problem by Graph Cut.

The second aspect we concentrated on, is the detection of facial points of interest, also called feature points. Where most of facial feature extraction techniques focus on either shape or appearance,we added anthropometric constraints to the Haar basis functions based Adaboost classification [46]. Those constraints permit to deal with all candidates as the appearance based classifier can not give only one result. Having the strength this results gave us, we extend the technique to the facial pose and feature position in 3D from a single image. To this end, we introduced prior constraints : the relative distances between 3D points, learned from a database. For facial features extraction as well as pose estimation from a single image, the problems are formulated as Markov Random Field involving pair-wise potentials and solve using Fast-PD, an efficient technique from linear programming to recover the lowest potentials.

The last chapter of the thesis presents the logical rest after face modeling and facial features extraction : Emotion Analysis. Taking advantages of the features we are now able to localize on a face, we proposed to model the 6 basic emotions (anger, disgust, fear, joy, sadness and surprise) through the displacement of those feature points in time. To this end, we used a Auto Regressive Process to estimate the position of points at time t from position of points at time $t-1$ [45, 46]. The new position are computed by feeding the input layer of a neural network with positions at the previous time. We also briefly introduced a novel approach of emotion recognition, based on the comparison between point positions prediction

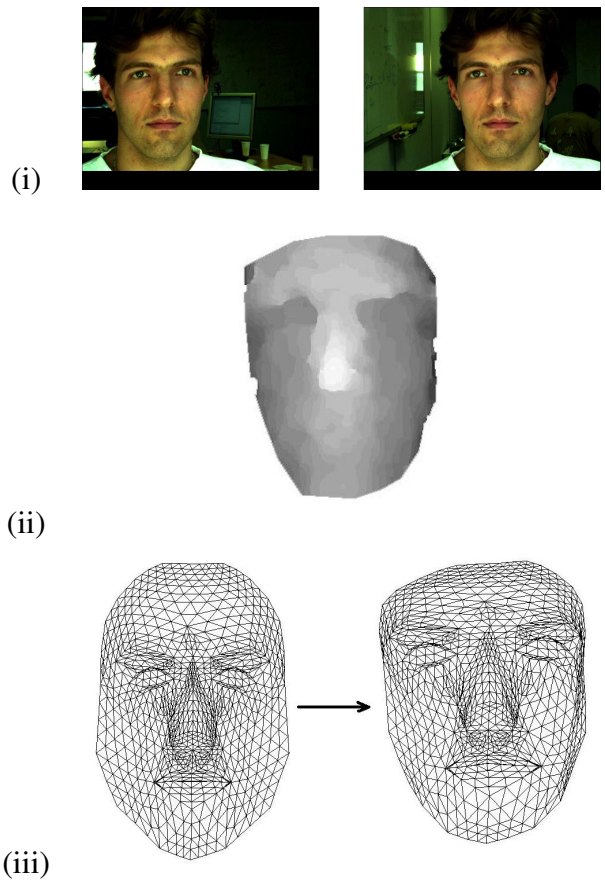


Figure 3.1: Reconstruction of a given face.

and actual point positions.

[Fig. (3.1)] shows a process involving parts of our work. From a pair of stereo images (i), the disparity map of the face is computed (ii). Then, our generic face model is registered on this reconstruction(iii). In [Fig. (3.2)], we present the resultant face reconstruction express joy through our an Auto Regressive Process defined in a learning step, while in [Fig. (3.3)] facial feature points are tracking in a sequence and their movements are reproduce on the reconstructed model.

For face reconstruction as well as for facial behaviour, the best technique can always be more realistic, as the perfect way to reproduce human behavior is not reached yet, and will probably not be in the next few years. Anyway, adding some tricks could give more realism to the animation. As already mentioned, the face is not perfectly symmetric. In the same way, even if, for a given expression, the points displacements look symmetric, they are actually not. By adding some per-

turbation, even to non moving points, the expression could appear more authentic. Another point to improve is the eyelids. In the mentioned techniques (state of the art methods, as mine), there is no particular consideration for the eyelids. However, one can assume that in the anger emotion, the frequency of flutter of the eyelids is faster than in fear or in surprise. But taking into account their speed and the frequency of a usual sequence, their detection and measurement is not feasible. Only post-processing addition of the eyelids movement can be possible. Realistic reconstruction and animation is not only a question of shape or movement. Appearance is really important too. Relighting a face [122] could help to express and emotion better, by making a smiling face lighter, or an angry person darker for example, or to embed a recontructed head in a new environnement.

Finally, we had to face an important problem : neither a 3D textured face database nor a 3D emotion database are available. In this context, it is really difficult to test the different method of face reconstruction, features detection or emotion recognition, particularly those based on a learning step. This is the reason why we created our own database, but with only 15 people. It also makes difficult the comparison when reviewing other researchers methods. Furthermore, when an emotion database exists, the expressions are fake. One can doubt the results of recognitions in real life. New researches on emotion recognition focus now on spontaneous emotions [119, 118, 4]. But the creation of a spontaneous expressions database is really a tough job, as, by definition, it is not possible to ask the subject to do one emotion or another, neither to tell him how to do it.

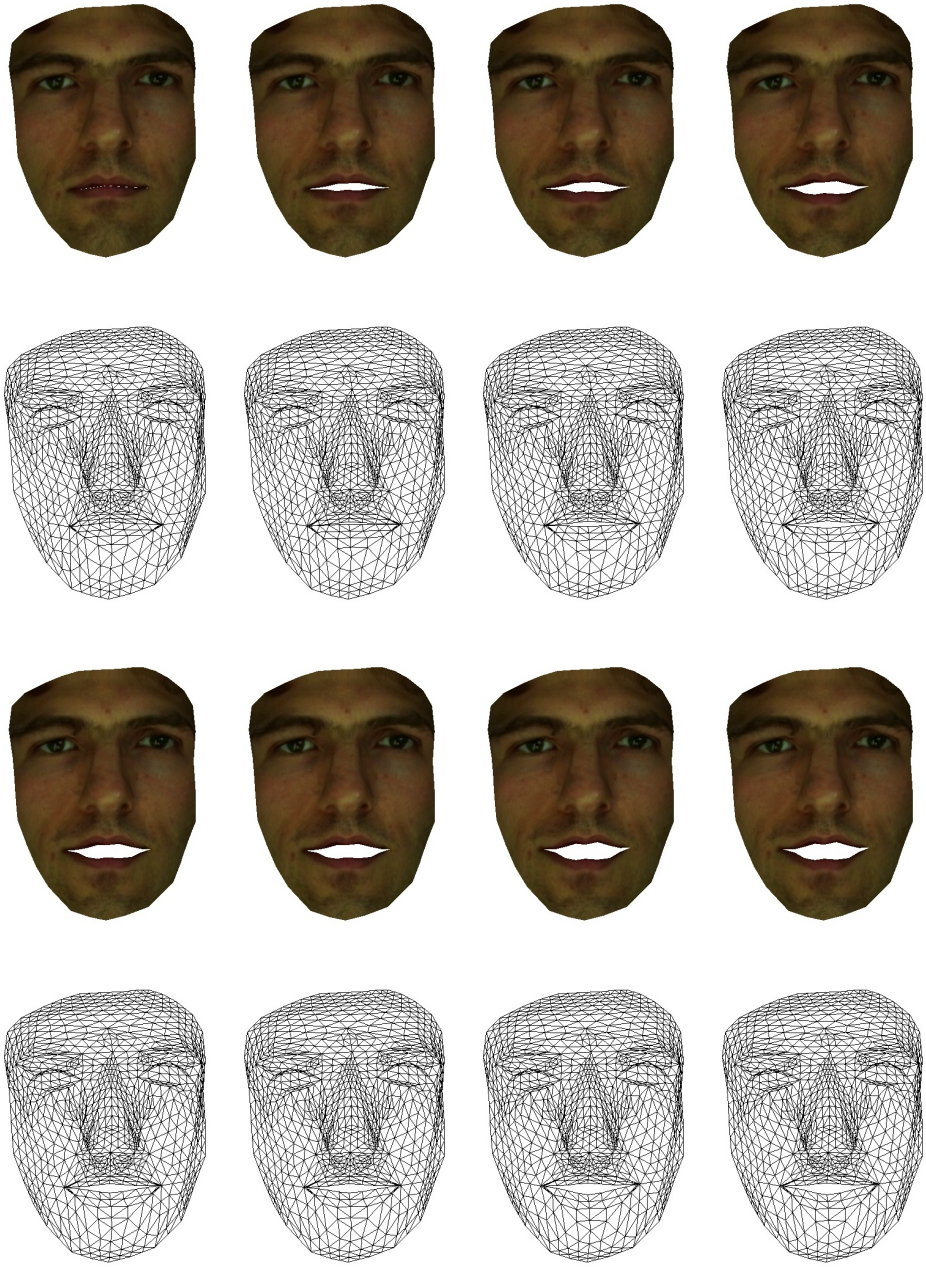


Figure 3.2: Joy learn and expressed by the reconstructed face.

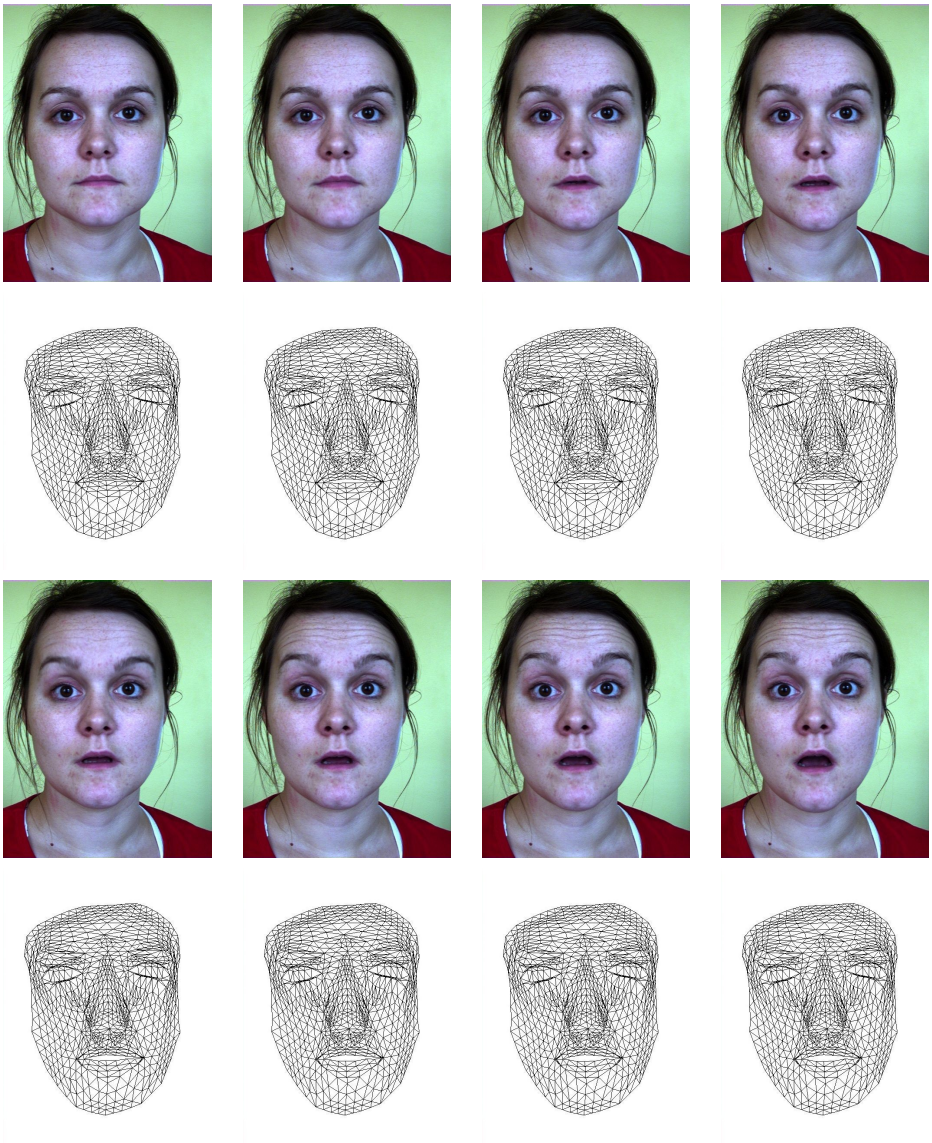


Figure 3.3: Mimicking of an expression using facial features tracking in a input sequence.

Conclusion (en français)

Dans cette thèse, nous avons abordé la plupart des domaines liés au visage en vision par ordinateur. Nous avons tout d'abord présenté un modèle de visage, sous la forme d'un maillage, dérivant du raffinement du modèle Candide, et de son recalage sur une base de données de reconstructions de visages 3D [46]. Ce modèle est suffisamment fin pour être réaliste et en même temps suffisamment simple pour l'adapter facilement à un sujet donné. Nous l'avons enrichi d'un ensemble de paramètres d'animation, définis par des points de contrôle, chacun étant associé à une zone d'influence. Le mouvement des sommets du maillage est alors calculé suivant le mouvement des points de contrôle associé à des fonctions radiales. En parallèle, nous proposons une nouvelle approche pour déterminer la géométrie 3D d'un visage, que nous avons appelé reconstruction super résolution [44], dont l'idée générale est de discrétiser l'intervalle de disparité à un demi pixel et de résoudre ce problème en utilisant les graph cut.

Le deuxième point sur lequel nous nous sommes attardés est la détection de points d'intérêt du visage. Là où la plupart des techniques d'extraction de points d'intérêt se focalisent sur la forme ou l'apparence, nous ajoutons des contraintes anthropométriques à une classification basée sur Adaboost [46]. Ces contraintes permettent de prendre en compte tous les candidats possibles retournés par une classification basée sur l'apparence. Forts de ces résultats, nous les avons étendus au cas de la pose 3D du visage à partir d'une seule image. Pour obtenir ce résultat, nous avons introduit des contraintes a priori : l'apprentissage des distances relatives entre les points 3D. Pour l'extraction des points d'intérêts, comme pour l'estimation de la pose du visage à partir d'une seule image, le problème est formulé suivant les champs de Markov, utilisant des pairs de potentiels, qui est résolu grâce à l'algorithme Fast-PD.

Le dernier chapitre de la thèse, en toute logique, après la modélisation du visage, est l'extraction des points d'intérêts, i.e. l'analyse d'émotion. Maintenant que nous étions capables de localiser précisément les points d'intérêts du visage, nous avons présenté la modélisation des 6 émotions de base (colère, dégoût, peur, joie, tristesse et surprise), en utilisant le déplacement de ces points d'intérêt dans le temps. Pour cela nous avons utilisé un processus auto-régressif pour estimer la

position des points au temps t à partir de la position des points au temps $t-1$ [45, 46]. La nouvelle position est calculée en alimentant la couche d'entrée d'un réseau de neurones avec les positions des points au temps précédent. Nous introduisons aussi brièvement une nouvelle approche pour la reconnaissance d'émotion, basée sur la comparaison de la position réelle des points et la position des points telle qu'elle est prédite par le modèle auto-régressif.

[Fig. (3.1)] nous montre un processus entier impliquant les différentes parties de notre travail. A partir d'une paire d'images stéréoscopique (i), la carte de disparité du visage est calculée (ii). Ensuite, notre modèle de visage est recalée sur cette recosntruction (iii). Dans la figure [Fig. (3.2)], nous présentons la reconstruction d'un visage exprimant la joie grâce à notre modèle auto-régressif. Enfin la figure [Fig. (3.3)] montre des points d'intérêts suivis dans une séquence et leurs mouvements répliqués sur le modèle reconstruit.

Pour la reconstruction du visage, aussi bien que pour les mouvements faciaux, la meilleure méthode peut toujours être plus réaliste, puisque le comportement humain ne peut pas encore être reproduit parfaitement, et ne le sera probablement pas avant plusieurs années. Néanmoins, ajouter quelques astuces pourrait donner plus de réalisme à l'animation. Comme nous l'avons déjà mentionné, le visage n'est pas parfaitement symétrique. De la même manière, même si pour une expression donnée, les déplacements des points paraissent symétriques, ils ne le sont pas réellement. En ajoutant de légères perturbations, mêmes aux points supposés immobiles, l'expression peut paraître plus authentique. Un autre point à améliorer serait les paupières, qui sont généralement mises de côté. Pourtant, on peut facilement supposer que pour une personne en colère, la fréquence de battement des paupières est plus importante que pour la joie ou la surprise. Mais compte-tenu de la vitesse de battements, le suivi est impossible, et seule une étape de post-traitement permettrait de les ajouter. Mais une reconstruction réaliste ne se base pas uniquement sur la forme ou le mouvement : l'apparence est très importante aussi. Eclairer un visage souriant ou assombrir un visage triste [122] pourrait aider à mieux exprimer une émotion.

Enfin, il est important de mentionner que ni une base de données de visages 3D texturés, ni une base de données d'émotions 3D n'étaient disponibles durant la thèse, ce qui nous a posé de sérieux problèmes, en particulier en matière d'apprentissage, et de validation de résultats que ce soit pour la reconstruction, la détection de points d'intérêts ou encore la reconnaissance d'expression. C'est pour cette raison que nous avons créé notre propre base de données, avec, hélas, seulement, 15 personnes. Pour les mêmes raisons, il nous a aussi été difficile de comparer nos résultats avec les autres méthodes de l'état de l'art. De plus, et c'est aussi le cas pour notre base de données, lorsqu'une base de données d'expressions existe, il s'agit le plus souvent d'expressions simulées. On peut alors douter des résultats des méthodes de reconnaissances d'expressions basées sur l'apprentissage et ap-

pliquées à la vie réelle. De nouveaux travaux sur la reconnaissance d'expressions se focalisent aujourd'hui sur les émotions réelles [119, 118, 4]. Mais la création d'une base de données d'émotions spontanées est un travail vraiment très difficile, puisque par définition, il n'est pas possible de demander au sujet d'exprimer telle ou telle émotion.

Appendix A

Anthropometric Constraints

Description of Acition Unitsare given in Tab. [A.1].

$(\mathbf{m}, \mathbf{n})$	Description	Constraints	$V_{\mathbf{m},\mathbf{n}}$
(0,1)	Outer corners of the eyes	$Symmetry$	$(0.y - 1.y)^2$
(2,3)	Inner corners of the eyes	$Symmetry$	$(2.y - 3.y)^2$
(5,6)	Inner corners of the eyebrows	$Symmetry$	$(5.y - 6.y)^2$
(7,8)	Outer corners of the eyebrows	$Symmetry$	$(7.y - 8.y)^2$
(9,10)	Upper midpoints of the eyes	$Symmetry$	$(9.y - 10.y)^2$
(11,12)	Lower midpoints of the eyes	$Symmetry$	$(11.y - 12.y)^2$
(13,14)	Nostrils	$Symmetry$	$(13.y - 14.y)^2$
(15,16)	Corners of the lips	$Symmetry$	$(15.y - 16.y)^2$
(9,10)	Midpoints of upper eyelids	$x = \frac{0.x+2.x}{2}$	$(9.y - 10.y)^2$ and $(9.x - \frac{0.x+2.x}{2})^2$ and $(10.x - \frac{0.x+2.x}{2})^2$
(11,12)	Midpoints of lower eyelids	$x = \frac{0.x+2.x}{2}$	$(11.y - 12.y)^2$ and $(11.x - \frac{0.x+2.x}{2})^2$ and $(12.x - \frac{0.x+2.x}{2})^2$
(17,18)	Midpoints of the lips	$x = \frac{15.x+16.x}{2}$	$(17.y - 18.y)^2$ and $(17.x - \frac{15.x+16.x}{2})^2$ and $(18.x - \frac{15.x+16.x}{2})^2$

Table A.1: MPEG-4 anthropometric constraints and corresponding pair-wise potentials.

Appendix B

Action Units of the Facial Action Coding System

Description of Acition Unitsare given in Tab. [B.1].

#	Description
1	Inner Brow Raiser
2	Outer Brow Raiser
4	Brow Lowerer
5	Upper Lid Raiser
6	Cheek Raiser
7	Lid Tightener
9	Nose Wrinkler
10	Upper Lip Raiser
11	Nasolabial Deepener
12	Lip Corner Puller
13	Cheek Puffer
14	Dimpler
15	Lip Corner Depressor
16	Lower Lip Depressor
17	Chin Raiser
18	Lip Puckerer
19	Tongue Out
20	Lip stretcher
21	Neck Tightener
22	Lip Funneler
23	Lip Tightener
24	Lip Pressor
25	Lips part
26	Jaw Drop
27	Mouth Stretch
28	Lip Suck
29	Jaw Thrust
...	

#	Description
...	
30	Jaw Sideways
31	Jaw Clencher
32	Lip Bite
33	Cheek Blow
34	Cheek Puff
35	Cheek Suck
36	Tongue Bulge
37	Lip Wipe
38	Nostril Dilator
39	Nostril Compressor
43	Eyes Closed
45	Blink
46	Wink
51	Head turn left
52	Head turn right
53	Head up
54	Head down
55	Head tilt left
56	Head tilt right
57	Head forward
58	Head back
61	Eyes turn left
62	Eyes turn right
63	Eyes up
64	Eyes down
65	Walleye
66	Cross-eye

Table B.1: Description of the Action Units of theFacial Action Coding System.

Bibliography

[1] M. Achroy and C. Beumier. The 3d_rma database.

[2] M. Alam, J. Bognar, R. Hardie, and B. Yasuda. Infrared Image Registration and High-Resolution Reconstruction Using Multiple Translationally Shifted Aliased Video Frames. In IEEE Transactions on instrumentation and measurement, volume 49, pages 185–203, 2000.

[3] C.S. Andersen. A survey of gloves for interaction with virtual worlds. Technical report, Aalborg University, Denmark, 1998.

[4] M. S. Bartlett, G. C. Littlewort, M. G. Frank, C. Lainscsek, and J. R. Fasel, I. R.and Movella1. Automatic recognition of facial actions in spontaneous expressions. Journal of Multimedia, 1:2049–2059, 2006.

[5] B. Bascle and A. Blake. Separability of pose and expression in facial tracing and animation. In International Conference on Computer Vision, pages 323–328, 1998.

[6] J.N. Bassili. Emotion recognition: The role of facial movement and the relative importance of upper and lower areas of the face. Journal of Personality and Social Psychology, 37:2049–2059, 1979.

[7] C. Basso, P. Paysan, and T. Vetter. Registration of expressions data using a 3d morphable model. In International Conference Automatic Face and Gesture Recognition, 2006.

[8] T. Beier and S. Neely. Feature-based image metamorphosis. In Siggraph, 1992.

[9] F. Bettinger and T.F. Cootes. A model of facial behaviour. In International Conf on Face and Gesture Recognition, 2004.

[10] Y. Black, M.J.and Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. International Journal of Computer Vision, 25:23–48, 1997.

[11] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In SIGGRAPH, 1999.

[12] F.L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. In Pattern Analysis and Machine Intelligence, 1989.

[13] G Borgefors. Distance transformations in digital images. In Computer Vision, Graphics, and Image, 1986.

[14] G. Box, G.M. Jenkins, and G.C. Reinsel. Time Series Analysis: Forecasting and Control. Prentice Hall, 1994.

[15] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. Pattern Analysis and Machine Intelligence, 26:1124–1137, 2004.

[16] Y. Boykov, O. Veksler, and R. Zabih. Efficient Approximate Energy Minimization via Graph Cuts. Pattern Analysis and Machine Intelligence, 20(12):1222–1239, 2001.

[17] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. Pattern Analysis and Machine Intelligence, 23(11):1222–1239, 2001.

[18] D. C. Brown. The bundle adjustment - progress and propects. International Archives Photogrammetry, 21, 1976.

[19] L. Chen, L. Zhang, H Zhang, and M. Abdel-Mottaleb. 3d shape constraint for facial feature localization using probabilistic-like output. In Automatic Face and Gesture Recognition, 2004.

[20] I. Cohen, N. Sebe, A. Garg, L.S Chen, and T.S. Huang. Facial expression recognition from video sequences: temporal and static modeling. Computer Vision and Image Understanding, 91:160–187, 2003.

[21] J.T. Connor, R. Douglas Martin, and L.E. Atlas. Recurrent neural networks and robust time series prediction. In IEEE Transactions on Neural Networks, 1994.

[22] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. Pattern Analysis and Machine Intelligence, pages 681 – 685, 2001.

[23] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In Pattern Analysis and Machine Intelligence, 2001.

[24] T.F. Cootes, D. Copper, C.J. Taylor, and J. Graham. Active shape models - their training and application. In Computer Vision and Image Understanding, pages 38–59, 1995.

[25] D. Cristinacce and T. Cootes. Facial Feature Detection using Adaboost with Shape Constraints. In British Machine Vision Conference, volume 1, pages 231–240, 2003.

[26] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In British Machine Vision Conference, 2006.

[27] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In CVPR, 1996.

[28] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. International Journal of Computer Vision, 38(2):99–127, 2000.

[29] D. DeCarlo, D. Metaxas, and M. Stone. An anthropometric face model using variational techniques. In Siggraph, 1998.

[30] G. Donato, M. S. Bartlett, and J. C. Hager. Classifying facial actions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21, 1999.

[31] S. Duffner and C. Garcia. A connexionist approach for robust and precise facial feature detection in complex scenes. In Fourth International Symposium on Image and Signal Processing and Analysis, 2005.

[32] P. Ekman and W.V. Friesen. Facial Action Coding System. Palo Alto, 1978.

[33] A. Essa and A. Pentland. Facial expression recognition using a dynamic model and motion energy. In ICCV, 1995.

[34] I. Essa. Coding, analysis, interpretation, and recognition of facial expressions. IEEE Trans. on Pattern Analysis and Machine Intelligence, 19, 1997.

[35] I.A. Essa and A. Pentland. A vision system for observing and extracting facial action parameters. In Computer Vision and Pattern Recognition, 1994.

[36] R. Feris, J. Gemmell, K. Toyama, and V. Krueger. Hierarchical wavelet networks for facial feature localization. In International Conference on Computer Vision's Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, 2001.

[37] D. Fidaleo and G. Medioni. Model-assisted 3d face reconstruction from video. In IEEE International Workshop on Analysis and Modeling of Faces and Gestures, 2007.

[38] R.A. Fisher. The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7:179–188, 1936.

[39] R. Forchheimer, I.S. Pandzic, and et al. MPEG-4 Facial Animation: the Standards, Implementations and Applications. John Wiley & Sons, 2002.

[40] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. Canadian Journal of Mathematics, pages 399–404, 1956.

[41] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. In Annals of Statistics, volume 28, pages 337–374, 2000.

[42] P. Fua. Using model-driven bundle-adjustment to model heads from raw video sequences. In International Conference on Computer Vision, pages 46–53, 1999.

[43] P. Fua and C. Miccio. Animated heads from ordinary images: A least-squares approach. Computer Vision and Image Understanding, 75(3):247–259, 1999.

[44] C. Ghys, N. Paragios, and B. Bascle. Graph-based multi-resolution temporal-based face reconstruction. In International Symposium on Visual Computing, 2006.

[45] C. Ghys, N. Paragios, and B. Bascle. Understanding 3d emotions through compact anthropometric autoregressive models. In International Symposium on Visual Computing, 2006.

[46] C. Ghys, M. Taron, N. Paragios, N. Komodakis, and B. Bascle. Expression mimicking : from 2d monocular sequences to 3d animations. In International Symposium on Visual Computing, 2007.

[47] A. Gunduz and H. Krim. Facial feature extraction using topological methods. In International Conference on Image Processing, 2003.

[48] H. Gunes and M. Piccardi. Fusing face and body display for bi-modal emotion recognition: Single frame analysis and multi-frame post integration. In International Conference on Affective Computing and Intelligent Interaction, 2005.

[49] H. Gupta, A. K. RoyChowdhury, and R. Chellappa. Contour-based 3d face modeling from a monocular video. In british Machine Vision Conference, 2004.

[50] M. Hennecke, K. Prasad, and D. Stork. Using deformable templates to infer visual speech dynamics. In Asimolar, 1994.

[51] B. Horn and B. Schunck. Determining Optical Flow. In Artificial Intelligence, volume 17, pages 185–203, 1981.

[52] R. Hsu, M. Abdel-Mottaleb, and A. Jain. Neural network-based face detection. Pattern Analysis and Machine Intelligence, 24(5):696–706, 2002.

[53] Y. Hu, B. Yin, and D. Kong. A new facial feature extraction method based on linear combination model. In International Conference on Web Intelligence, 2003.

[54] X. Huang, N. Paragios, and D. Metaxas. Establishing local correspondences towards compact representations of anatomical structures. In MICCAI, 2003.

[55] X. Huang, N. Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. In Pattern Analysis and Machine Intelligence, 2006.

[56] X. Huang, S. Zhang, Y. Wang, D. Metaxas, and D. Samaras. A hierarchical framework for high resolution facial expression tracking. In Workshop on Articulated and Nonrigid Motion, 2004.

[57] M. Isard and A. Blake. Condensation, conditional density propagation for visual tracking. International Journal of Computer Vision, 29(2):5–28, 1998.

[58] M. Isard and A. Blake. Unifying low-level and high-level tracking in a stochastic framework. In European Conferrence on Computer Vision, 1998.

[59] P. Kalra, A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. 3d interactive free form deformations for facial expressions. In Compugraphics, 1991.

[60] T. Kanade, J.F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In Automatic Face and Gesture Recognition, 2000.

[61] H. Kang, T.F. Cootes, and C.J. Taylor. Face expression detection and synthesis using statistical models of appearance. In Measuring Behavior, 2002.

[62] M. Kass, A. Witkin, , and D. Terzopoulos. Snakes: Active contour models. In International Conference on Computer Vision, 1987.

[63] V. Kolmogorov and M. Wainwright. On the optimality of tree-reweighted max-product message passing. In Conference on Uncertainty in Artificial Intelligence, 2005.

[64] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? Pattern Analysis and Machine Intelligence, 26(2):147–159, 2004.

[65] N. Komodakis and G. Tziritas. Approximate labeling via graph-cuts based on linear programming. In Pattern Analysis and Machine Intelligence, 2007.

[66] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. International Journal of Computer Vision, pages 199 – 218, 2000.

[67] A. Lanitis, C.J. Taylor, and T.F. Cootes. Automatic interpretation and coding of faces images using flexible models. PAMI, 19(7):743–756, 1997.

[68] Dimitris Samaras Lei Zhang, Sen Wang. Face synthesis and recognition under arbitrary unknown lighting using a spherical harmonic basis morphable model. In IEEE International Conference on Computer Vision and Pattern Recognition, 2005.

[69] R. Lengagne, P. Fua, and O. Monga. 3-d stereo reconstruction of human faces driven by differential constraints image and vision computing. Image and Vision Computing, Special Issue on Facial Image Analysis, pages 337–343, March 2000.

[70] W.-K. Liao and I. Cohe. Classifying facial gestures in presence of head motion. In CVPR, 2005.

[71] W.-K. Liao and I. Cohen. Belief propagation driven method for facial gestures recognition in presence of occlusions. In CVPR, 2006.

[72] J.L. Lien, T. kanade, J.F. Cohn, and C.-C. Li. Automated facial expression recognition based on facs action units. In IEEE International Conference on Automatic Face and Gesture Recognition, pages 390–395, 1998.

[73] Z. Liu, Z. , Zhang, C. Jacobs, and M. Cohen. Rapid modeling of animated faces from video. In International Conference on Visual Computing, 2005.

[74] Z. Liu, Y. Shan, and Z. Zhang. Expressive expression mapping with ratio images. In SIGGRAPH, 2001.

[75] M.H. Mahoor, M. Abdel-Mottaleb, and A.-N. Ansari. Improved active shape model for facial feature extraction in color images. In Journal of Multimedia, 2006.

[76] M. Malciu and F. Prêteux. Tracking facial features in video sequences using a deformable model-based approach. In Conference on Mathematical Modeing, Estimation and Imaging, volume 4121, August 2000.

[77] T. Molet, R. Boulic, and D. Thalmann. A real time anatomical converter for human motion capture. In Eurographics Workshop on Animation and Simulation, 1996.

[78] P. Mordohai and G. Medioni. Stereo using monocular cues within the tensor voting framework. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28:968–982, 2006.

[79] J. Noh, D. Fidaleo, and U. Neumann. Animated deformations with radial basis functions. In ACM Symposium on Virtual Reality Software and Technology, 2000.

[80] T. Otsuka and J. Ohya. Recognizing multiple persons facial expressions using HMM based on automatic extraction of significant frames from image sequences. In in Proc. International Conf. on Image Processing, 1997.

[81] C. Padgett and G. Cottrell. Representing face images for emotion classification. Advances in Neural Information Processing Systems, 9, 1997.

[82] M. Pantic and I. Patras. Dynamics of facial expression: Recognition of facial actions and their temporal segments from face profile image sequences. IEEE Transactions on Systems, Man and Cybernetics, 36:433–449, 2006.

[83] M. Pantic and L.J.M. Rothkrantz. Expert system for automatic analysis of facial expression. Image and Vision Computing Journal, pages 881–905, August 2000.

[84] S. Park, M. Park, and M. Kang. Super-Resolution Image Reconstruction : A Technical Overview. In IEEE Signal Processing Magazine, pages 21–36, 2003.

[85] D.I. Parke. Computer generated animation of faces. In AMC National Conference, 1972.

[86] i. Patras and M. Pantic. Particle filtering with factorized likelihoods for tracking facial features. In IEEE International Conference on Automatic Face and Gesture RecognitionWorkshop on Human Computer Interaction, pages 97–102, 2004.

[87] P Penev and J. Atick. Local feature analysis: A general statistical theory for object representation. Neural Systems, 1996.

[88] P. Perez, C. Hue, J. Vermaak, and M Gangnet. Color-based probabilistic tracking. In European Conference on Computer Vision, 2002.

[89] M.K. Pitt and N. Shepard. Filetring via simulation : auxiliary particle filtering. American Statistical Association, 94:590–599, 1999.

[90] S. M. Platt and N. I. Badler. Animating facial expression. In Siggraph, 1981.

[91] J.-P. Pons, R. Keriven, and O. Fageras. Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score. International Journal of Computer Vision, 72(2):179–193, 2007.

[92] J. P. Pons, R. Keriven, O. Faugeras, and G. Hermosillo. Variational stereovision and 3d scene flow estimation with statistical similarity measures. In International Conference On Computer Vision, pages 597–602, october 2003.

[93] M. Rosenblum, T. Yacoob, and L. Davis. Human expression recognition from motion using a radial basis function network architecture. IEEE Transaction on Neural Network, 7:1121–1138, 1996.

[94] M. Rydfalk. Candide, a parameterized face. Technical report, Dept. of Electrical Engineering, Linköping University, 1987.

[95] Y. Saatci and C. Town. Cascaded classification of gender and facial expression using active appearance models. In International Conference on Automatic Face and Gesture Recognition, 2006.

[96] J. A. Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, June 1999.

[97] Y. Shan, Z. Liu, and Z. Zhang. Model-based bundle adjustment with application to face modeling. In International Conference on Computer Vision, 2001.

[98] H. Tao and T.S. Huang. Connected vibrations : A modal analysis approach to non-rigid motion tracking. In CVPR, 1988.

[99] Y. Li. Tian, T. Kanade, and J Cohn. Recognizing action units for facial expression analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2):97 – 115, 2001.

[100] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. International Journal of Computer Vision, 9:137–154, 1992.

[101] M.F. Valstar and M. Pantic. Combined support vector machines and hidden markov models for modeling facial action temporal dynamics. In IEEE Workshop on Human Computer Interaction, 2007.

[102] M.F. Valstar, I Patras, and M. Pantic. Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data. In CVPR, volume 5, pages 76–84, 2005.

[103] P. Viola and M.J. Jones. Robust real-time face detection. International Journal of Computer Vision, 2004.

[104] D. Vukadinovic and M. Pantic. Fully Automatic Facial Feature Point Detection Using Gabor Feature based Boosted Classifiers. In IEEE Conference on Systems, Man and Cybernetics, 2005.

[105] C.L.Y. Wang and D.R. Forsey. Langwidere : A new facial animation system. In Computer Animation, 1994.

[106] H. Wang and N Ahuja. Facial expression decomposition. In ICCV, 2003.

[107] S. Wang, L. Zhang, and D. Samaras. Face reconstruction across different poses and arbitrary illumination conditions. In Biometric Authentication Workshop, pages 91–101, 2005.

[108] K. Waters. A muscle model for animating threedimensional facial expression. Computer Graphics, July 1987.

[109] X. Wei, Z. Zhu, L. Yin, and Q. Ji. A real time face tracking and animation system. In Computer Vision and Pattern Recognition Workshop, 2004.

[110] J. Whitehill and C. O. Omlin. Haar features for facs au recognition. In 7th International Conference on Automatic Face and Gesture Recognition, 2006.

[111] Y. Wu and T. Huang. A co-inference approach to robust tracking. In International Conference on Computer Vision, 2001.

[112] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In Conference on Computer Vision and Pattern Recognition, 2004.

[113] R. Xiao, L. Zhu, and H.J. Zhang. Boosting chain learning for object detection. In International Conference on Computer Vision, 2003.

[114] T. Yacoob and L. Davis. Recognizing human facial expressions from long image sequences using optical flow. Pattern Analysis and Machine Intelligence, 18:636–642, 1996.

[115] P. Yang, Q. Liu, and D.N. Metaxas. Boosting coded dynamic features for facial action units and facial expression recognition. In CVPR, 2007.

[116] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. Advances in Neural Information Processing Systems, pages 689–695, 2000.

[117] A. Yezzi, S. Soatto, H. Jin, A. Tsai, and A. Willsky. Mumford-Shah for Segmentation and Stereo., chapter 12, pages 207–227. Springer, 2003.

[118] Z. Zeng, Y. Fu, G.I. Roisman, Z. Wen, Y. Hu, and T.S. Huang. One-class classification on spontaneous facial expression. In International Conference on Automatic Face and Gesture Recognition, 2006.

[119] Z. Zeng, M. Pantic, G.I. Roisman, and T.S. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. IEEE Transaction on Pattern Analysis and Machine, 2007.

[120] L. Zhang and D. Samaras. Face recognition under variable lighting using harmonic image exemplars. In IEEE International Conference on Computer Vision and Pattern Recognition, 2003.

[121] L. Zhang and D. Samaras. Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics. Pattern Analysis Machine Intelligence, 28(3), 2006.

[122] L. Zhang, Y. Wang, S. Wang, and D. Samaras. Image-driven re-targeting and relighting of facial expressions. In Computer Graphics International, pages 11–18, 2005.

[123] Z Zhang. Feature-based facial expression recognition: Sensitivity analysis and experiments with a multilayer perceptron. International Journal of Pattern Recognition and Artificial Intelligence, 13, 1999.

[124] F. Zuo and P de With. Fast facial feature extraction using a deformable shape model with haar-wavelet based local texture attributes. In International Conference on Image Processing, 2004.