Università degli Studi
di Milano

Dottorato di Ricerca in Informatica
XIX Ciclo
Settore Scientifico Disciplinare INF/01 Informatica

Extension of Active Appearance Models for improved face analysis and expression retrieval

Tesi di Dottorato di Ricerca di:
Julien PEYRAS

Relatore:
Prof. Patrice DALLE

Correlatore:
Prof. Paola CAMPADELLI

Coordinatore del Dottorato:
Prof. Vincenzo PIURI

Anno Accademico 2007/08

Abstract

The field of *Computer Vision* has known great improvements and many tools, always more powerful, have been proposed to face number of difficult problems aiming to automatically understand the content of images and videos.

A special branch called *Face Processing* focusses in particular on the analysis of images and videos presenting faces.

Among all the possible applications, we find the interesting fields of Human-Machine Interaction, driver attention surveillance, medical or psychological studies.

A tool for automatical extraction of facial deformations on videos has been the object of intense investigations for about two decades now. The development of such a tool presents many difficulties because of the extreme variability of faces, as well as the high number of possible facial deformations. Moreover, they are often accompanied with head pose changes and not homogeneous, or even changing illumination, what further complexify the task of extraction of the relevant information from the pictures.

Two approaches have revealed to be more promising to face this complex problem.

The first one, for which main advances we owe to the University of San Diego, is based on the machine-learning theory. Fast classification tools analyse the picture or a subpart of it and declare the presence or absence of a particular facial deformation. They work fast and robustly, but do not provide accurate spatial information on the face structure and way it deforms. This can be a limitation for some applications in the medical field for instance.

The second approach, called Active Appearance Models (AAMs) and initially proposed by Tim Cootes *et al.* in 1998, is based on a statistical modelling of facial shape and appearance possible variations. A large amount of contributions around the AAMs can be attributed to the Carnegie-Mellon University. The model tends to adapt its shape and appearance in order to best match the face present on a picture. If this match, commonly called *fitting*, is performed correctly, some post-processing systems can exploit the information of shape and appearance to interpret them.

The quality of the interpretation strongly depends on the fitting accuracy of the model AAM on the face present on the picture. In the literature, AAMs have been reliably used in a context called *person-specific* where the tracked face is also statistically learnt by the AAM. The most generic case of fitting on unseen faces (faces not learnt in the model's statistics) still present many problems, and AAM presents many limitations that not always are clearly presented in the existing works, often because they are not well known from many AAM users.

In this thesis we propose to deeper investigate the AAM solution and to understand what they are really capable of. We reach a better understanding of the way to use the AAMs at their maximum performance, and we also propose some solutions to extend their capabilities. Whereas the recent years of research have not given the AAM the convincing credibility that it could be a reliable solution to the most generic problem of fitting on unseen faces displaying various expressions, under varying pose and changing light, we bring new hopes that it can eventually be.

The thesis is organized as follows.

1. Chapter 1 presents the facial analysis problem and its interests, as well as the state of the art in the field of automatical retrieval of facial expressions. A special attention is given to AAMs and all similar methods that were proposed in the literature.
2. Chapter 2 presents the construction rules of an AAM, and the one possible optimization process used to align it onto a face present on an input picture, namely the *Simultaneous Inverse Compositional* algorithm proposed by Matthews and Baker.

3. The beginning of chapter 3 presents the various possible contexts of use of an AAM, and shows when it works well, and when it doesn't. The remaining of the chapter investigates the most favourable case, the *person-specific* context, in order to understand its limitations. This study shows that the more specialized the AAM, the higher its performance, even in the person-specific context. This gives us an idea on the way the AAM fitting on unseen faces can be maximized. It is maximized when the less information is learnt by the AAM: basically, nothing else than the identity should vary.
4. We thus investigate the case of AAM training and fitting on identity varying faces, with expression, pose and illumination fixed. In chapter 4 we give an original explanation of the limitations of the fitting accuracy on faces that are unseen from the model, and an extension to perform highly accurate face fitting: we make use of specialized and segmented models to increase the fitting accuracy on unseen faces in the restricted context of frontal and neutral faces illuminated homogeneously. The fitting accuracy obtained is often comparable to statistical manual labellings accuracy on the tested images. This is a strong contribution since the fitting on unseen faces is still an open problem today, and many applications depend on such an ability.
5. In chapter 5 we extend the concept of model specialization to face the problem of varying pose and expression, always on unseen faces. We present some preliminary results to the proposed solution. For the small expressive face database we built and used, the results are very promising. Further improvements could clearly be obtained if more identities were included in the database, and if we could have a higher control on the quality and intensity of the expressions displayed: the exactness of the pose and expression performed by each participant of the database is crucial for the quality of the results.
6. Chapter 6 presents an original solution to deal with light variations. The AAM is basically unrobust to illumination changes. We present a new formulation that we call *Light-Invariant AAM* or *LI-AAM*. The proposed optimization framework projects the 3 dimensional color data into a 1 dimensional light-invariant space where the training data can be aligned with the input picture data despite their not corresponding illumination.
7. Chapter 7 finally presents a solution to segment the pixels of a face that are hair, or glasses, and substitute them with their equivalent non-hair, or non-glasses pixels, basing the reconstruction on the observation of the remaining of the face in order to harmonize the reconstruction with this face. The solution is based on the use of different appearance statistics (taken from the AAM paradigm) containing or not faces with facial hair or glasses. This solution could find applications in people recognition or in re-styling for instance.

To Francesca

Acknowledgement

I would like to thank Paola Campadelli for offering me the opportunity to lead this PhD within the State University of Milan. I particularly thank the people from LAIV, Stella, Elena, Sonia, Stefano, Gabriele, Giuseppe, for all the good time I had with them, and for there support through the difficult moments. Thanks to all my friends and to all the dear people from Milan and other parts of Italy, thanks for being so fantastic and for the great moments I spent with them.

I specially thank Patrice Dalle, and all the members if the TCI research team from IRIT-Toulouse for making me feeling so well among them, and for giving me the opportunity to attend and present my work to many conferences. Obviously I feel extremely grateful to Hugo Mercier for the great collaboration we had together, our priceless discussions leading to interesting and original ideas and tests, and for the hard work and support we brought one another during a year and a half. A special thought goes for all my friends and best friends from Toulouse, they will know who they are!

Thanks to Ivan and Mathieu, my eternel friends.

Thanks to the people from LASMEA and particularly Vincent Gay-Bellile and Mathieu Perriollat for welcoming me so well within the lab and for the productive exchanges we had. Thank you to my friend and very talented collaborator Daniel Pizarro, I hope I will carry on the research on some interesting topics with you.

Big thanks to all the great people from FittingBox and youarethemodel.com who I spend a great time with. Thank you especially to Ariel Choukroun, for the motivation and positivity he insuflates, and for our great scientifical partnership.

Thank you very much to Marian Stewart Bartlett, Simon Baker, and Adrien Bartoli for kindly reviewing my PhD work. Special thanks to you Adrien for being so thorough, consciencious with me and my work on faces you were initially not particularly into. Thank you for making me secure in the choices and research directions I took eventually. I have learnt a lot with you. Thank you so much.

Thank you infinitely to my Mum for her infite love and sincerity. I really think you could not bring me more than you did. I love you.

Great thanks go to my father who is great, and who we all love despite his great flaws I wish he could analyse and commonly admit one day. You really helped me in many difficult situations of my life Dad, and I am so grateful and proud for that.

Thanks to my sister Lorène, the best, I don't know what I would do without you! You're simply the best!

The best of my feelings and gratitude goes for Francesca. You, who have loved me and supported me during all the bad and great times. You are my stability, my light, my will to laugh, my joy.
I love you so much Fra.

Contents

Chapter 1

The Facial Expression Analysis Problem

1.1 Face, rich source of information

The problem of recognizing emotions from facial expression observation has attracted much interest of people from various disciplines and for a long time. In 1649, the phylosopher René Descartes [38] introduced the six *simple passions*: Wonder, Love, Hatred, Desire, Smile and Sadness and assumes that all others are composed of some of these six. In 1872, the biologist Charles Darwin [37] argued that there are specific inborn emotions and that each emotion includes a specific pattern of activation of the facial expression and behavior. Inspired from these works of Darwin, Ekman, Friesen and Ellsworth [44, 43] investigated on the universality of the expression of six basic emotional expressions namely *Surprise*, *Anger*, *Disgust*, *Happiness*, *Sadness* and *Fear*.

More than a simple support for six categorized emotions, the face considered either at passive or dynamic state is a rich vector of information and for mankind stands as the first mean in interpersonal communication. A face at passive state conveys two kind of information:

- The face (we consider non-pathological cases) has an intrinsic structure in correspondence to morphological properties presented by human race. However, among the billions people on earth, each face is different since it possesses its own characteristics. We can say that a face bony structure and tissues always present certain characteristics that are common to all faces, but each particular face presents a certain anthropomorphic instance in structure and dimension of these common components, making it unique. Particularities on an individual's face allow to identify the person among others (the face recognition branch of face processing is interested in this),
- Someone's face also conveys information on the person's personality, attractiveness, age and gender.

Facial activities can convey three kinds of information:

- Facial signals (often combined to head movement, also known as pose change) can convey information on someone's emotional state (ranging much more than the six universal emotions), intentions or cognitive activity,
- Facial signals can convey co-verbal information. In interpersonal interactions, Mehrabian [3] indicated that in communication, spoken words of a message contributes only for 7% of the message as a whole, the vocal intonation contributes for 38% while facial expression of the

speaker contributes for 55% to the spoken message. This implies that the facial expressions form the major modality in human communication,

- Lip movements accompanying the spoken message is an element that can be used for strengthening the verbal understanding. During communication it often occurs together with facial expressions possibly making it difficult to decorrelate the two signals.

Observing that the human face plays a major role in communication, it has become the object of studies for disciplines like face-focussed medical fields, psychology, linguistic or cognitive and behavioral science fields. In behavioral science, giving an interpretation to some facial activity have led to many studies, only within restricted contexts however. For instance, in a study on posed and spontaneous smiles, it has been identified that self-report enjoyment was correlated with the simultaneous occurrence of orbicularis oculi and zygomatic major actions, whereas smiles featuring only zygomatic major were observed for posed expression [41]. In the same way, genuine and fake pain can be differenciated by observing or not a particular facial activity [35]. Person telling the truth or lying also can be differenciated through accurate facial observation [51]. Facial activity can predict the onset and remission of depression, schizophrenia, and other psychopathologies [45], can discriminate suicidally from non-suicidally depressed patients [61], can predict transient myocardial ischemia in coronary patients [93], and can present particular patterns involved in alcohol intoxication [96].

1.2 Coding the facial signals

These studies have required a thorough observation of the facial activity. Researchers trained to observe the faintest facial movements have found interesting correlations between signals and inner states, whereas non expert people usually fail to note the signals that help to see the difference.

The ideal thing is then to establish an objective coding system that lists all possible facial events prone to vehicle information (in reality they are all bearer of information). This coding system can then allow the expert to know what are the possible events, and then to pay attention to their occurrence. There mainly exist two coding systems: the one used for MPEG4 compression, and another one called FACS that better suits the behavioral science community.

The MPEG4 standard [100] specifies the face in its neutral state with a set of feature points: the Facial Definition Parameter set (FDPs). The tracking of these feature points aim to provide a Facial Animation Parameter set (FAPs). MPEG4 standard aims at defining facial expressions as a set of measurements (FDPs) and transformations (FAPs). This standard is useful for video compression but it lacks of subtlety for finer facial analysis. Some subtil facial deformations such as cheek raiser activated by the orbicularis oculi are not taken into account by this standard although it can discriminate between two similar ways for expressing joy.

The Facial Action Coding System (FACS) [42] from Ekman and Friesen (1978) is probably the most known study on facial activity. It is a system that has been developed to facilitate objective measurement of facial activity for behavioral science investigations of the face. FACS is designed for human observers to detect independent subtle changes in facial appearance caused by contractions of the facial muscles. In a form of rules, FACS provides a linguistic description of all possible, visually detectable, facial changes in terms of 44 so-called Action Units (AUs). Using these rules, a trained human FACS coder decomposes a shown expression into the specific AUs that describe the expression.

When a trained coder has labelled a video sequence, all facial action occurring along the video frames should be extracted and labelled and all the useful information should be available for studies or interpretation:

- behavioral scientists can perform studies establishing correlations between some inner state and some coexisting facial events/actions.

- interpreters can use the results of the latter community to obtain information concerning the inner state of the person by observing his facial activity that has been decomposed into Action Units.

Since FACS pretends to be the most comprehensive coding system, other attempts to extract the facial signals and to interpret them present the risk to miss some facial information that we know will be taken into account by the FACS. Coding the facial signals with FACS is becoming a widely adopted convention since it has proved to effectively report all the richness of facial information and to be the most useful coding method for behavioral studies.

FACS encompasses:

- changes in the head pose
- deformation of intransient features: eyebrows, eyes, nose (though being less deformable than others, nostrils external parts are rather subject to displacements) and mouth.
- transient features occurrence such as wrinkles that accompany the deformation of intransient features. A distinction must be made between permanent and transient wrinkles that can either make appearing new visual elements or strengthen already existing permanent wrinkles. Typically, the amount of transient wrinkles is obtained by difference between current wrinkle intensity and the intensity of intransient wrinkles captured at rest (neutral face).

But it does not include two points that could be added to the coding system to make it complete. Indeed, facial event can also occur without a physical displacement.

- chemical or hormonal process creating a reaction/appearance variation on the face, like when someone's face turns to red.
- occurrence of new elements, like tears shedding or tongue pulled out.

For now, current advances in facial activity understanding has remained quite restrained. The task of interpreting biological signals still needs to be intensively investigated by the behavioral science community. One of the main obstacles to doing research on emotion is the lack of a fast, reliable and automatic tool for extracting facial events [40, 52]. Current analysis are still operated manually on pictures aquired from a video camera. This often requires to operate a very long, tedious and expensive manual labelling work prone to lack accuracy when boredom arises. FACS coding is currently performed by trained experts who make perceptual judgments of video sequences, often frame by frame. It requires approximately 100 hours to train a person to make these judgments reliably and pass a standardized test for reliability. It then typically takes over two hours to code comprehensively one minute of video. The experts can be subject to tiredness or boredom what can increase the risk that coding pieces lack of accuracy. Furthermore, although humans can be trained to code reliably the morphology of facial expressions (which muscles are active) it is very difficult for them to code the dynamics of the expression (the activation and movement patterns of the muscles as a function of time). There is good evidence suggesting that such expression dynamics, not just morphology, may provide important information [43]. For example, spontaneous expressions have a fast and smooth onset, with distinct facial actions peaking simultaneously, whereas posed expressions tend to have slow and jerky onsets, and the actions typically do not peak simultaneously [53].

The apparition of an automatic tool for accurate facial analysis would find a huge amount of interesting applications. As a first one we find telecommunication where bandwidth requirement is always more important and current investigations aim at reducing transmitted data through always smarter compression methods: this is why the MPEG4 standard was invented. In [22], new face image tracking and coding is developed to compress facial expression information. Another field of interest is *human-machine interaction* where we find a recent field of activity known as *affective computing*

[89] aims at improving the dialog quality between the machine and its user. It usually involves a multi-modal perceptive architecture to perceive and interpret all messages from the user that could convey an information concerning his internal state, his intents or his verbal and non-verbal message within an improved communication framework. This interesting and ambitious research field combines approaches in the main three modalities: vision with facial expression extraction that motivates the work presented here, auditive with speech understanding and kinesthesic with corporal activity usually obtained through sensors applied on the body or inserted in clothes, in a *wearable computing* fashion. A subsequent treatment aims at merging all signals together in order to give them a relevant interpretation and to infer some information about the mental state of the observed person. Pantic and Rothkrantz [84] develop the topic and give the basic rules for building an affect-sensitive multimodal human-computer interface.

In the face-focussed medical domain, we attend a large growing interest for such an extraction tool in order to automate some spatial and temporal facial event measurements so allowing a faster and more accurate analysis. Current analysis are still operated manually on pictures aquired from a video camera. This often requires to operate a very tedious manual labelling work prone to lack of accuracy when boredom arises and that could be dismissed with the introduction of an automatic computer vision tool.

Automatic facial analysis also finds applications in video-conference, facial animation for infographists, video-surveillance, biometric access systems, fatigue detection for drivers *etc.*

1.3 Existing work on automated facial events extraction

Recent advances in image analysis and pattern recognition open up the possibility of automatic detection and classification of emotional and conversational facial signals without being intrusive. Automating facial expression analysis could bring facial expressions into man-machine interaction as a new modality and make the interaction tighter and more efficient. Such a system could also make classification of facial expressions widely accessible as a tool for research in behavioral science and medicine. Behavioral studies could be considerably speeded-up and new interpretation rules could efficiently be discovered and used for building smart interfaces.

Another approach called electromyography (EMG) can retrieve the facial signals. However this technique requires to place sensors on the face, which may inhibit certain facial actions and which rules out its use for naturalistic observation.

In this work, we propose to develop a tool based on computer vision. In the following we describe the context we work in and constraints dealt with.

1.3.1 Constraints and context

We want to develop a tool based on computer vision that is useful for facial analysis and in particular facial expression analysis. Three main fields are interesting for us:

- advanced human-machine interfaces able to detect and interpret facial cues occurrence,
- behavioral research where scientists crave new ergonomic tools that would help them push forward their studies,
- medical field where people is interested in having objective measures of the facial characteristics and dynamic.

We want to build a tool that is suitable for these fields. Considerating the kind of applications we want to set up, the constraints we will have to respect should be in priority:

- **accuracy**, which is necessary to allow more reliability for a post-treatment stage of interpretation,
- **automaticity**; we want the system to be as automatic as possible (human intervention should be inexistent or extremely non-tedious),
- **generalizability**; we would also like the system to be able to deal with any kind of person,
- **speed**, since we would like the system to tend toward real-time processing,
- **robustness** to light changes that are very likely to happen in everyday situations.

We might tolerate a system that is still not perfect for treatment on people presenting particularly difficult face (large hair occlusion for instance). Since aimed applications should run in everyday computer use context, the system should be able to deal with head poses that are prone to occur: it should be able to handle facial motion extraction in context of frontal, as well as limited out-of-plane rotation of the head.

1.3.2 Related work

During the past two decades, a number of methods have appeared in the computer vision literature for facial expression analysis.

Variety of approaches have been observed, including optical flow, tracking of high level features, methods based on statistical learning of images, and methods based on biologically inspired models of human vision. Pantic [82] (2000) as well as Fasel and Luettin [47] (2003) have reviewed existing approaches and have clearly understood the problem of expression analysis and recognition. A more up-to-date survey would be of great use soon.

1.3.2.1 Optical flow based approaches

Optical flow methods aim at computing the direction and magnitude of motion along video frames. For face analysis, optical flow have been used to extract facial actions by recovering muscle activity or displacement of precise points of interest on the face. Mase and Pentland [74] detected the direction of optical flow over the movement-deformed facial surface. They were perhaps the first to track action units using optical flow. Although their method was simple, the results were sufficiently good to show the usefulness of optical flow for observing facial motion.

Yacoob and Davis [114, 115] used optical flow to represent facial motion and recognize the six universal facial expressions introduced by Ekman. Intransient feature are assumed to be given in the first frame and are tracked in subsequent frames. Optical flow is estimated between the current and the first frame, giving then a symbolic representation of facial changes. Eventually, this representation is classified into one of the six facial expressions.

Rosenblum, Yacoob and Davis [94] expanded the previous work with a radial basis function neural network for each one of the six universal facial expressions.

Black and Yacoob [19, 20] improved the robustness to head motion with local parameterized model of image motion for facial expression analysis. Rigid head motion is represented by a planar model that covers and track the whole face. Inner facial feature motions are represented by an affine plus curvature model. Initial region of head and facial features are selected manually in the first frame.

Essa and Pentland [46] have proposed the combination of a dynamic physical model and of motion energy for facial expressions classification. Motion is estimated from optical flow and is refined by the physical model in a recursive estimation and a control framework. A physical face model is applied to model facial muscle actuation and an ideal 2D motion is computed for five studied expressions similar to the universal facial expressions. Each template have been delimited by averaging the patterns of

motion generated by two subjects for each expressions. Facial expressions classification is based on the Euclidean distance between the learned template of motion energy and of the observed image.

Cohn *et al.* [23, 25] proposed an automatic recognition system based on the AUs modeling. The displacement of 36 manually located feature points are estimated using optical flow. Separate groups were used for the classification of the AUs. They used two discriminent functions for three AUs of the eyebrow region, two discriminant functions for three AUs of the eye region and five discriminant functions for nine AUs of the nose and mouth regions.

Lien *et al.* [69] propose an hybrid method based on: first, feature point tracking is performed using a coarse-to-fine pyramid method, second, Lucas-Kanade dense flow tracking [70] combined with principal component analysis (PCA), third, high gradient component analysis in the spatio-temporal domain to extract expression information. Expression classification is based on FACS action units. Hidden Markov Models (HMMs) are used to discriminate between AUs or combination of them according to the pattern of feature motions. The method can work for frontal views only.

Major drawbacks of the method rely in that image quality are often required to be very high since noise can considerably mislead the flow estimation. Variation of illumination due to light or due to varying reflectances accompanying facial deformation also influence negatively the optical flow. Although Black and Yacoob proposed a solution to cope with rigid motion as well as non-rigid deformations, it is difficult to accurately decorrelate both information and to retrieve the feature motion private from rigid movement.

1.3.2.2 Detection or tracking of high-level feature - Feature modeling templates or contours

In this approach, facial features are modeled by geometrical features or templates either rigid or deformable. Each template or feature is given the possibility to displace by 2D similarity and possibly to modify its shape to better match the feature it has been designed to fit onto. The displacement and shape variation parameters are usually optimized through the minimization of a cost function into which those parameters are embedded. The doctoral theses [60, 68] make use of such deformable templates.

Tian *et al.* [103] use two separate Neural Networks (NNs) based approach to recognize 6 upper and 10 lower AUs based on both transient and intransient facial features (deepening of facial furrows and wrinkles). The facial features have been grouped into separate collections of feature parameters because they claim that the facial actions in the upper and lower face are relatively independent for AU recognition [42]. Note that Whitehill *et al.* [107, 108] show that they are not and study their co-occurrence. The inputs of the NNs for both training and classification are the parametric descriptions of permanent and transient facial features. The facial features are manually initialize in the first frame and tracked in the remaining frames of the sequence. The facial expression recognition is realized by the combination of the upper face and the lower face AUs.

Pantic and Rothkrantz [83] use face models made of dual-view points for facial expressions classification: the frontal view and the side view. After automatic segmentation of facial features through multiple feature detection techniques, they code several points of interest into AUs using a set of rules. Then the FACS is used to recognize the six universal facial expressions. The classification is performed by comparing the AU-coded description of facial expressions of observed expression against the rule descriptors FACS. The feature detection techniques might be old-performing and it is not sure whether they can compete with state-of-the-art statistical based feature detection methods.

Pantic and Patras [81] recently proposed an automatic system for analysing large amount of facial actions on profile views. Particle filtering (see Isard and Blake [62]) is used to track 15 points of interest and temporal rules are used to recognize the facial actions. Dealing with profile view can be

interesting but the challenge first consists in tackling the problem of non-frontal view ranging 0 to up to 45 degrees out-of-plane rotation for building non-intrusive human-machine interfaces.

Cohn, Kanade *et al.* [24] built an automated facial image analysis (AFA) system that make use of several different kind of methods. The head is detected automatically with the Rowley-Baluja-Kanade neural network based face detector [95]. Automatic and robust recovery of 3D-head motion is performed through Xiao's [111] 3D tracking by means of a cylinder taken as rough approximation of the head structure and able to robustly track the head rotation in monocular video sequences. A reference template composed of pixels of the image region covered by the cylinder surface is extracted in the first frame. In subsequent frames, the 6 degrees of freedom (d.o.f.) of the cylinder are optimized to minimize the difference between the reference template and the current region covered by the cylinder. Head motion is then recovered automatically and used to warp (or stabilize) the face image to canonical view. Features are detected and tracked (offline) throughout the sequence. Feature extraction is obtained through Active Appearance Model fitting [75] or through Active Shape Model [118]. To track or analyse permanent (intransient) facial features, a blend of techniques are used such as optical flow mentioned before with Lien *et al.* [69] work, gabor wavelets from Tian *et al.* [102] that have shown good ability to discriminate eye region action units, multi-state models offering the possibility to switch from a model to other, each being more specialized to deal with a given configuration of a feature. For instance, mouth is represented for open, closed, or tightly closed configuration. Indeed, optical flow does work well with qualitative changes involving new visual element appearance or disappearance. A generative model fitting approach for eye state analysis [79] presented by Moriyama, Xiao *et al.* is used for modeling interpersonal variation of the eyes. Transient facial features are localized thanks to the prior knowledge on intransient feature location. Transient features are extracted through gabor wavelet or edge detection techniques. A set of feature dynamic parameters, as feature displacement, velocity, and appearance, are retrieved as well as head motion trajectories, and this information is given to a classifier for action unit recognition. Timing of action units is also available. Facial features are divided into two groups of parameters: the upper and the lower faces that are almost completely independent with regard to AU extraction. Neural networks (with one layer hidden) are used for classification. The author highlight the importance for the features (visual information) to be measured precisely and have high specificity for the target action units. This multi-module complete approach has been tested for approximately 20 action units and presents high agreement with manual FACS coding. On frontal pose and given lighting condition, average recognition accuracy exceeded 93% on unknown subjects (test subjects different from subjects used for training) displaying posed expression and for AU occurring singly. Spontaneous behavior with slight head rotation is a more difficult task to face. Also, it is a current active research topic. This reference work represents the accomplishment of about ten year work of a large group of researchers.

1.3.2.3 Biologically inspired approaches

A number of methods have been inspired by the human visual system. The eye is known to be sensitive to gradient and spatiotemporal variation of visual elements in the scene. Based on this observation, researchers have found interesting to evaluate the capability of gradient extractor features like wavelets (see Mallat [72]) to represent the useful information conveyed by the face.

Zhang [116] as well as Zhang, Lyons *et al.* [117] compared features geometry and gabor wavelets response for their capability to correctly discriminate expressions of various emotions (always the six universal emotions [44] plus the neutral state). A gabor wavelet is nothing more than a 2D sine wave that is weigthed with a gaussian bell: its response is given by a coefficient that indicates the similarity degree between the image region it lies on and the length and orientation of the wave. The geometry

as well as the wavelet location are defined through manual selection of points of interest on images that are manually aligned and cropped. Both geometrical information and gabor responses are given as input to a multilayer perceptron trained for expression recognition. Recognition rate has shown to be much higher when using gabor wavelets, and further improvement is achieved when combining both geometric positions and gabor responses. This study was done on homogeneous subjects including only Japanese and it aimed at recognize only emotion-specified expressions.

Tian *et al.* [102] operated a similar test and they highlighted two interesting things. The first is the good performances achieved with gabor wavelets for single AU recognition, what is consistent with previous results. The second is the surprisingly poor recognition results obtained for gabor responses compared to geometrical information when studying AU combinations, non-homogeneous subjects, or small head motion. Another difference with previous work is that images are not preprocessed. These differences suggest that any advantage of gabor wavelets in facial expression recognition may depend on manual preprocessing and may fail to generalize to heterogeneous subjects and more varied facial expression. Nevertheless, [117] and [102] coincide in the fact that best recognition rate is achieved when geometrical and appearance based features are combined. Finally combining a bunch of visual cues extractors seems a good solution to improve recognition.

Bartlett, Littlewort, Lainscek, Fasel and Movellan [14, 16, 17] proposed a solution for facial expression and AUs recognition based on machine learning methods. The method is fully automatic: face detection is performed on each frame, the cropped region of the face is resized to be convolved with a bank of gabor energy filters, and then passed to a recognition engine of different kind. Engines based on adaboost, Support Vector Machines (SVM), linear discriminant analysis (LDA), and feature selection classifiers are tested. Best filters are selected with adaboost, and the response of those filters is provided to SVMs. Universal expressions are recognized as well as 18 Action Units of the FACS description with very high percentage. Moreover, the margin of SVM answer indicates the dynamic of facial actions: intensity and time information of the beginning, evolution, appex and release of the action can be observed through the SVM margin evolution.

This team bases its approach on machine learning methods, distinguishing themselves from feature extraction and analysis where extraction accuracy conditions the good expression analysis. Here, the developed system can be invariant to small inaccuracies in face cropping since the classifying engines also learn examples that are not perfectly aligned. Illumination changes (maybe rather small) can also be handled for the same reason. The overall simplicity and good performance of this solution makes it extremely fascinating and also a reference work.

The following and more recent works of this team [15, 13] is very promising for tackling the problem of AUs recognition and intensity analysis in video sequences where expressions are displayed spontaneously. Spontaneous expression are harder to analyse than played (or posed) expressions since they are often weaker and accompanied by head rotations.

Their face and center of eyes detector [48] based on the real-time haar-feature like object detector introduced by Viola and Jones [106] has kindly been made available online[1]. We will use this face and eye detector in the last chapters of this thesis.

1.3.2.4 Statistical learning approaches

In facial feature extraction and analysis, the quality and accuracy of the extraction is the crucial condition for their good forthcoming analysis. Basically, image appearance (and thus facial feature appearance) are used to automatically retrieve the feature shape. To extend briefly on the discussion, let's say that the appearance information (mainly gradient and visual contrasts) are critical to find

[1] http://mplab.ucsd.edu/projects-home/project1/free-software/MPTWebSite/API/index.html

the right feature shape. When the appearance deviates from what the fitting algorithm is designed to see, the shape fitting is likely to be wrong (and this is the achilles' heel of most deformable template methods). Since image acquisition and identity are highly variable parameters, a well performing fitting method should be able to cope with them, also when they occur unexpectedly. To this end, a great deal of research has been devoted to develop statistical approaches. In a general manner, number of possible data are learnt in advance, allowing to deal with novel examples, somehow close to the learnt data.

A large quantity of statistical tools and representation have been tested and used for face analysis in general. In [12] Bartlett *et al.* tested the efficiency of many learning techniques for the purpose of FACS encoding. Surveyed unsupervised techniques are principal component analysis (PCA), local feature analysis (LFA) and independent component analysis (ICA). Supervised techniques are Fisher's Linear Discriminant (FLD). This work consists in collecting a serie of face pictures of subjects displaying various Action Units as well as the neutral expression. For each picture onto which a subject performs a facial action, the aligned neutral picture of the same subject is subtracted. The data obtained in this way is used for training, except the one being tested (this is called leave-one-out test). Statistical representation method are used in turn, and their capability for classifying the test data is tested. For one method, one data representation is built on data corresponding to the same facial action (for varying identities). The test data is consecutively projected onto each facial action space, and it is associated to the space that best models it. This similarity is measured in terms of euclidean distance and cosine of the angle between feature vectors. Gabor wavelets discriminancy power was also tested. ICA and gabor wavelets proved to best discriminate between the 12 AUs that were tested. According to previously stated Tian *et al.* [102], the good gabor wavelet discriminability power accounts for the nature and quality of the used pictures.

Lanitis, Taylor and Cootes [67] (2007) makes use of an Active Shape Model (ASM), statistically modeling the range of shapes a face can assume. The shape is modeled by landmarks manually placed on a training set of faces, and a shape analysis and principal component analysis (PCA) is performed on all training shapes to infer the ASM, a statistical model that is fitted onto test faces thank to an optimization algorithm presented in [32, 31]. Once the face is fitted, the underlying pixels describing the face appearance is extracted and projected onto an appearance space to retrieve some parameters that synthesize the appearance information. Shape and appearance parameters are used to recover 3D pose, identity, gender, and to recognize the expression of the face. Simple correspondences or classification rules are set up for the purpose. 3D head pose is recovered with only the shape information. Two on all shape parameters are used to evaluate the 3D head pose angulation: authors claim that the first and third parameters are responsible for appearant changes of pose. In reality there is no insurance for this since each parameter describes a part of the training set shape variance without associating a special semantical meaning to it. This variance associated to a parameter completely depends on the training set and on images it contains. Usually first two parameters gathering major variance describe most of the dominant variation attributed to pose change, but there is no exact rule for it. This assertion is unreliable. Identity is retrieved by associating the shape and appearance parameters (in combination or separately) to the mahalanobis-closest cluster of identity lying in a representation space where several individuals are represented under various pose and expressions. It is not clear whether test is performed on images different from images used to train the classifier. The expression recognition is also achieved using shape and appearance parameters. Training is performed on images representing the universal basic expressions: a different class was built for each expression and test images are associated to one of the classes. The presented techniques are promising but their use lack of care and experiments are done without a complete understanding and control of the model behavior.

1.3.2.5 Comments

From the above state-of-the-art, the first point that comes out is the importance to retrieve the visual information accurately. The more accuracy we reach in measures, the best can be the classification into AUs.

Two leading works seem to pave the way of research in automatic facial expression analysis. The first called AFA [24] (an related works) is proposed by the people from Carnegie Mellon University (CMU) led by Jeffrey Cohn and Takeo Kanade presents a coarse to fine approach. Face is detected, features are extracted as accurately as possible, head movements are tracked and compensated, and features are followed accurately. The set of extracted visual and geometrical information is passed to classification. The drawback of the method might come from the head motion compensation. The rotated face is warped thanks to a cylindrical approximation of the head. This operation might introduce distorsion on the face visual content. The CMU's group approach presents an advantage for us: they perform an accurate feature extraction and tracking. This is useful for medical applications and accurate static and dynamic description of the face. The second reference work called Machine Perception Toolbox or MPT box (pronounce *empty*) is based on learning machine techniques specialized in the detection of a particular element. The face is detected first, then the eyes, and features areas and explored with detectors specialized in one AU recognition. This effective approach is also probably more robust than any method based on accurate facial feature extraction. However this extraction is one of our requirements.

We will therefore concentrate on the first approach. After face is detected, facial feature are extracted by means of Zhou's Active Shape Model fitting method [118] (2003) or by Matthews and Baker's Active Appearance Model fitting method [75] (2004).

Statistically modeling shape and appearance of faces as done with AAMs seems a complete and promising approach: retrieving both the shape and the appearance information from a face can easily lead to almost any analysis on that face and its current configuration (and enlightment). The Matthews and Baker solution for fitting an AAM onto an image is also very attractive due to its appealing mathematical formulation and the face fitting accuracy they show on demos and results. Some more clues (appeared over the last two years: the decision to follow this strategy was already taken) are also pretty convincing. Authors have not stopped to publish a large amount of interesting papers based on their initial implementation, and new impulses are also coming from De La Torre *et al.* [66, 55]. The AAM approach basically deals with 2D shape deformation and appearance variation phenomenom, but they re-created the 3D motion effect (a 3D face is simply considered as its reprojection onto 2D). 3D movements such as head pose change can apparently be treated with this method. The *inverse compositional* workframe it is based on makes of it an efficient AAM fitting process, contrarily to 3D approaches that are usually very slow. As a consequence, many arguments bring us onto the *inverse compositional AAM fitting algorithm* proposed by Matthews and Baker.

Active Appearance Models used under the Matthews and Baker's fitting framework can suit the requirements of our target applications. Before getting into concrete exploration of an AAM structure and this efficient fitting procedure, we will have another look at the literature. We will present the historical background that conducted to Active Appearance Model invention. In the other hand, AAM and derivated methods have been for ten years an active source of research and it is important to be aware of some of the works that gravitate around this method. This second tour of the art is proposed in the following.

1.3.3 Work related to the Active Appearance Models

Statistical methods applied for face analysis started in the early beginning of the nineties, with Kirby and Sirovich [63] and Turk and Pentland [105] who applied a principal component analysis (PCA) on

facial images. PCA is a linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. The coordinates obtained by PCA are orthogonal. PCA can be used for dimensionality reduction in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. Facial images can be encoded in low-dimensional PCA subspace and can be reconstructed in the original image space, which is useful for video-compression. The problem of eigenfaces is that reconstructed images are blurry. This effect happens because the PCA is performed on face images that are not properly aligned semantically, and image elements does not fall into exact correspondence. It results that the low-dimensional PCA subspace is unable to compensate for the high image frequencies introduced by wrong alignments and reconstructed picture looks blurry.

First efforts to compensate the wrong semantical alignment appeared with Craw and Cameron [36]. They sample face textures by setting few manual landmarks on each face image. A PCA is applied on these textures reprojected onto a reference frame. Cootes *et al.* [26, 30] proposed a more thorough representation of face shape. They also placed landmarks onto face images, but they set them in semantical correspondence across faces. The landmarks forming a configuration on each face defined the shape of that face. The PCA is then applied on these configurations where points are defined in correspondence.

The algorithm to fit the statistical face shape representation onto an image is the so called ASM [29] proposed by Cootes and Taylor in 1992. The shape models is translated, rotated, scaled and deformed along its shape parameters to be placed into correspondence with higher gradients in the image. The problem of this method was its limited robustness to face image variability since strong edges do not always represent the same part of the face. A second statistical aspect was introduced to cope with this limit. In [33], ASM modelled the grey-level pixel value along the normal of the contour at each model point. During the search process, a sequence of point locations are tested within a distance to each model point. The new position reached at each iteration is chosen to minimize the error reconstruction between the local grey-level profile and the grey-level modelled normally to that point. Of course, grey-level models of points on the face side are for a half defined outside the face area, in the background that can be highly variable and source of perturbation for the ASM fitting process.

Further investigation was undertaken to develop performing methods on this basis. Rather than using face texture only locally along the normal of shape contour on some points, two main approaches appeared using the complete face texture information. Sclaroff and Isidoro [97] proposed Active Blobs in 1996, and Cootes and Taylor [27] introduced the Active Appearance Model in 1998.

An Active Appearance Model or AAM is basically composed of the statistical shape and a statistical appearance. AAM eventually unifies the eigenfaces paradigm with the shape alignment workframe properly introduced with the ASM. Face textures are sampled inside the shape formed by landmarks onto each image. A warping process allows to express these textures in the same reference frame where alignment is ensured (up to noise in label placement). A PCA is applied on these texture data to obtain another PCA subspace, this one statistically models face textures. More details on AAM structure and construction will be given in next chapter.

The AAM fitting process is a non-linear optimization problem usually solved by iteratively updating the shape and appearance parameters of the model. The aim is to minimize the sum of square difference between the image modelled by the AAM and the input face image. In [27] an additive update is performed through gradient descent. For each parameter involved in the optimization one gradient (gradient image) is used. The gradients are computed by numerical differentiation, generating small

variations on each parameter. This operation requires to define an amplitude coefficient for generating the perturbation of each parameter what is critical for the overall system performances. Moreover, for computation savings the gradients are left constant along the fitting process which is generally incorrect. The gradients generally depends on the current parameters and should then be recomputed at each iteration. The approximations made in [28] lead to poor performances in terms of both fitting accuracy of the final fit, and the number of iterations required to converge.

Alternative methods have been proposed to address the problem of AAM fitting like [59] where AAMs are fit to the image within a particular filtering framework bringing improved robustness to illumination changes.

Many authors inspired their research on the AAM paradigm. In turn, it is interesting for us to explore their work and find the inspiration to elaborate our new solutions. We propose to review three authors.

1.3.3.1 Ahlberg's works

Ahlberg [2] presented a near real-time 3D model-based method attractive for its simplicity. The 3D model used (CANDIDE [1]) can model the face identity by optimizing some parameters that conditions the shape aspect linked to the identity, and it can also optimize another set of parameters dedicated to the expressivity of the person. The method suffers from many drawbacks:

- the mathematical optimization formulation that considers constant the jacobi matrix when rigorously speaking it is not and it should be recomputed at each iteration. Moreover, the gradients are computed by numerical differentiation, generating small variations on each parameter. This operation is critical since it requires the determination of an amplitude coefficient used for generating the perturbation of each parameter what is a very constraining task.

- the manual intervention necessary to initially adjust the model to the person face: the identity parameters does not seem to be optimized automatically.

- the size of the reprojection patch that is very small, probably in order to save computation resources that highly depends on this parameter. This leads to downsampling the input image and lose information on it, what should have a direct negative influence on the fitting accuracy.

- Each coded Action Unit is represented with only one deformation vector. This might be a too simple modelling of AUs and might result in a loss of accuracy if we consider the high variability to express even one single AU among different people. However, if robustness is a priority this approach might be suitable, although the lack of fitting accuracy usually induces a lack a robustness too. Considering the subtlety of facial deformation we need to deal with, accuracy is a high priority for us, and we cannot rely on such a coarse face and expression encoding.

In brief, both the mathematical optimization formulation and the simplicity of the combined approach identity/expressions seem to finally lead to a not satisfactorily accurate system. Moreover, a consistant manual training and instantiation of the identity shape is necessary, thus not fulfiling the requirement of automaticity.

1.3.3.2 Blanz and Vetter *et al.*

Blanz and Vetter [21] proposed a dense 3D model solution which seems to reach amazing accuracy but relies on a too heavy face statistics and process to fit it. Romdhani and Vetter [92] implemented the Baker's efficient inverse compositional image alignment algorithm smartly adapted to fit a 3D model onto 2D pictures, resulting in speed increase. The system is still far from processing in real-time. Moreover, important manual intervention seems to be necessary since the fitting process is very likely to get stuck into local minimas (the huge dimensionality of the statistics used makes the fitting task

extremely complex).

1.3.3.3 Muñoz *et al.*

Muñoz *et al.* [80] (2005) use a Structure From Motion (SFM) factorization method in order to build a 3D deformable model of the person to track in the video sequence. Once the model is available, they make use of the efficient inverse compositional image alignment algorithm to fit it onto the video sequence. The method is very interesting but can only be used when a model of the person is available and a reliable method to build one automatically is not available yet.

1.3.4 Comments and direction

The overviewed existing methods all focus on the particular task of intransient feature deformation extraction. To deal with transient feature extraction requires to set up further method that focus on the precise area of occurrence. Please note that it should not be too difficult to localize these areas relatively to the already tracked intransient features, and then to analyse them with the appropriate tool, task that we leave as future work. Our aim will be to localize the facial intransient features (eyebrows, eyes and mouth), and to measure the direction and intensity of displacement. We decide to investigate the AAM potentiality to help us to reach that goal.

Chapter 2

Active Appearance Models (AAMs) and Image Alignment Algorithms

AAM have been the object of many investigations for almost one decade now. It has become one of the most fashionable techniques used for face analysis. This method smartly combines a variable appearance with a deformable shape in order to model faces and their possible variability. A *fitting process* is generally used to align the model onto a face represented on an input picture.

Initially introduced by Cootes and Edwards[1] in [28], these authors fit the model onto an image using a numerical optimization method that makes use of several approximations. The first one is the numerical computation of the gradient images that are obtained by finite differencing the gradients of the reference template image along each deformation component. Some background pixels are included in the pair of images they make a difference on. This is a pain because the background can change when the head position changes on subsequent frames. They introduce a trick [28] to lessen the effect of background pixels but this does not lead the gradients to be optimally computed anyway. The differenciation step should be set manually and is a critical parameter highly influencing the fitting speed and accuracy. The second approximation stands in the fact that these gradient images are left constant through iterations: unfortunately in a *forward additive parameter optimization process* (presented in details at 2.2.1) like the one used in [28], these gradients depend on the current parameters and should then be recomputed at each iteration. These two approximations result in a non-optimal performance of the AAM fitting speed and accuracy.

In [75, 8] Matthews and Baker presented a novel AAM fitting framework where higher performance is achieved in terms of speed and accuracy, due to both a rigorous way to compute the gradients, and the use of a novel efficient alignment algorithm. The image gradients are computed analytically, and they are recomputed at each iteration *when this is necessary*. Indeed in [8], they also introduced a novel fitting method that allows the optimization process to precompute the gradients once and to optimize the parameters in a search space where gradients do not depend on these parameters. This novel algorithm called *Inverse Compositional Image Alignment Algorithm* was presented in [9] and differs from the forward additive algorithm for what it reverses the role of the input image and the reference template image that must be aligned: instead of iteratively refining the alignment of the input image on the template image, this is the template onto which the refined alignment is estimated to better match the input image; but this estimate refinement is not directly applied to the template image; instead its inverse quantity is composed to the current estimate on the input image.

[1] Please note that the original terminology chosen by Cootes *et al.* refer to appearance as the combination of shape and texture. Here, we use the term appearance for the texture, and do not use any concise term to refer to their combination, we simply call it *shape and appearance*.

One curious thing to notice is how the original AAM fitting method introduced in [28] has been successful and nowadays keeps attracting most researchers when their results could be noticeably improved if they used more rigorously computed gradients. We explain the past and present popularity of Cootes' AAM (building and) fitting solution for it is available online, whereas the rareness of use of the Baker's solution could be due to the fact that the implementation of this solution is not straightforward following the steps proposed in [75].

In the following we present the way AAMs are built in 2.1, then both important algorithms for image alignment, the forward additive and the inverse compositional will be presented in 2.2.1 and 2.2.2. Some steps of the implementation of the AAM fitting process using the inverse compositional algorithm are described in 2.3. The only small difference we introduced with respect to the implementation proposed in [75] comes from the way we compute the current estimate parameters.

2.1 AAM construction

An Active Appearance Model describes an object of a predefined class as an instance of shape and an instance of appearance. Each object, for a given class, can be represented by its shape, described by vertex coordinates and an appearance, described by pixel intensities. It is defined by:

1. a shape $\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} p_i \boldsymbol{s}_i$, where $\boldsymbol{s}_0$ is the mean shape, $\boldsymbol{s}_i$ are shape deformation vectors and p_i are coefficients that weight the deformations;
2. an appearance $\boldsymbol{A}(\boldsymbol{x}) = \boldsymbol{A}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(\boldsymbol{x})$, where $\boldsymbol{A}_0(\boldsymbol{x})$ is the mean appearance image, $\boldsymbol{A}_i(\boldsymbol{x})$ are the appearance variation vectors and λ_i are their weighting coefficients.

Now let's see how to build this object, linearily morphable in 2D shape and appearance. A statistical analysis is successively operated on the shape of various training images belonging to a training set, and on the appearance of the same (or different) set of images.

2.1.1 The shape component

The first step in AAM construction consists in building a *shape space* that describes in a convenient manner a set of shape data. In a general manner, statistical shape analysis (a complete discussion can be found in [39]) aims at analysing the shape of different objects extracted from a same population and that have the same overall structure, but are all different one from others due to proper particularities in their respective morphology. Studied objects are generally purely 2D or purely 3D objects, or they can also be 3D objects reprojected in 2D on a still picture as it is the case for us: 2D pictures representing faces will be used.

Shape representation. To characterize in shape a set of N training examples coming from a same population (faces in our case), a set of landmarks must be defined for each example. A landmark is a point that must be chosen to place into correspondence the same semantically defined point recognizable on all objects of the training collection. There exist three kinds of landmarks:

- *anatomical landmarks* that are points assigned by an expert that correspond between organisms in some biological way. E.g. corner of an eye if objects are faces.
- *mathematical landmarks* that are points located on an object according to some mathematical or geometrical property of the figure. E.g. at a point of high curvature or at an extreme point.
- *pseudo-landmarks* that are constructed points on an organism, located either around the outline or in between anatomical or mathematical landmarks.

The sequence of landmarks that is defined on each training object is called a *configuration of points*, or *configuration of landmarks* (or also *vertices*). Consequently to the definition of a landmark (it must indicate the same location among the set of objects coming from the same population), all object configurations are described with an equal number of landmarks.

If each configuration C_i is composed of v landmarks that are defined by d coordinates (we consider the studied object in a d dimensional space, where $d = 2$ in the case of images), the configuration C_i can be represented by a vd element vector formed by concatenating the coordinates of each individual landmark.

For instance for a two dimensional configuration C_i for which the landmark l is represented by two coordinates as so (x_{il}, y_{il}), the $2v$ element vector $\boldsymbol{V_{C_i}}$ representing the configuration C_i is:

$$\boldsymbol{V_{C_i}} = (x_{i1}, ..., x_{il}, ..., x_{iv}, y_{i1}, ..., y_{il}, ..., y_{iv})^T \tag{2.1}$$

Shape normalization. Given all the training examples expressed both by a set of landmarks in their original d dimensional space and by their associated vector expressed in nd dimensions, we proceed to extract the shape from these examples. Following the definition given by [39], the shape is all the geometrical information that remains when location, scale and rotation effects are filtered out from an object. Indeed, the statistical shape analysis has sense only if the considered shape data are represented in the same co-ordinate frame. We wish to remove variation which could be attributable to the allowed transformation. This can be achieved by a Procrustean analysis operated on the configurations. An example can be followed on Figure 2.1 on faces 2D represented on 2D pictures and where landmarks have been placed manually. First, the location is removed and barycenters are placed onto the origin of the coordinates where configurations are defined (2D or 3D). Second, each configuration scale is adjusted in such a way that the vector composed of the stacked coordinates of the landmarks is of norm one. Third, each centered and normalized configuration is iteratively rotated in order to minimize the sum of euclidean distances that separate the corresponding landmarks between each configuration and a mean configuration (see Figure 2.1)).

Given N training configurations, the shape retrieval can be formalized as follows:

(1) Compute the centroïd coordinates for each configuration C_i (for $d = 2$, $(\overline{x}_i, \overline{y}_i) = \left(\frac{1}{v}\sum_{l=1}^{v} x_{il}, \frac{1}{v}\sum_{l=1}^{v} y_{il}\right)$).

(2) Compute all centered configuration vectors $\boldsymbol{V'_{C_i}}$ (for $d = 2$, $\boldsymbol{V'_{C_i}} = (x_{i1} - \overline{x}_i, ..., x_{iv} - \overline{x}_i, y_{i1} - \overline{y}_i, ..., y_{iv} - \overline{y}_i)^T$).

(3) Choose one example i as initial estimate of the mean shape and scale this example so that its norm equals 1: $\boldsymbol{s}_0 = \frac{\boldsymbol{V'_{C_i}}}{||\boldsymbol{V'_{C_i}}||}$.

Iterate (4) and (5) until the mean shape does not change significantly after an iteration:

(4) Align all the shapes with the current mean estimate $\boldsymbol{s}_0$ to obtain the shapes $\boldsymbol{s}_i$ so as to minimize $\sum |\boldsymbol{s}_i - \boldsymbol{s}_0|^2$.

(5) Re-estimate $\boldsymbol{s}_0$ from aligned shapes: $\boldsymbol{s}_0 = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{s}_i$, and scale $\boldsymbol{s}_0$ to norm 1.

(6) When the mean estimate suits our stability criterion, we project the shapes $\boldsymbol{s}_i$ into the tangent space to linearize the shape data around the mean shape $\boldsymbol{s}_0$. This is done by scaling each $\boldsymbol{s}_i$ by $1/(\boldsymbol{s}_i^T.\boldsymbol{s}_0)$

Step (4) consists in aligning each shape onto the mean shape. When two shapes $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ have been translated in order to place their centroïd on the origin, the alignment of a shape onto the other is obtained by fixing the first and scaling and orientating the second in order to minimize $\sum |\boldsymbol{s}_1 - \boldsymbol{s}_2|^2$,

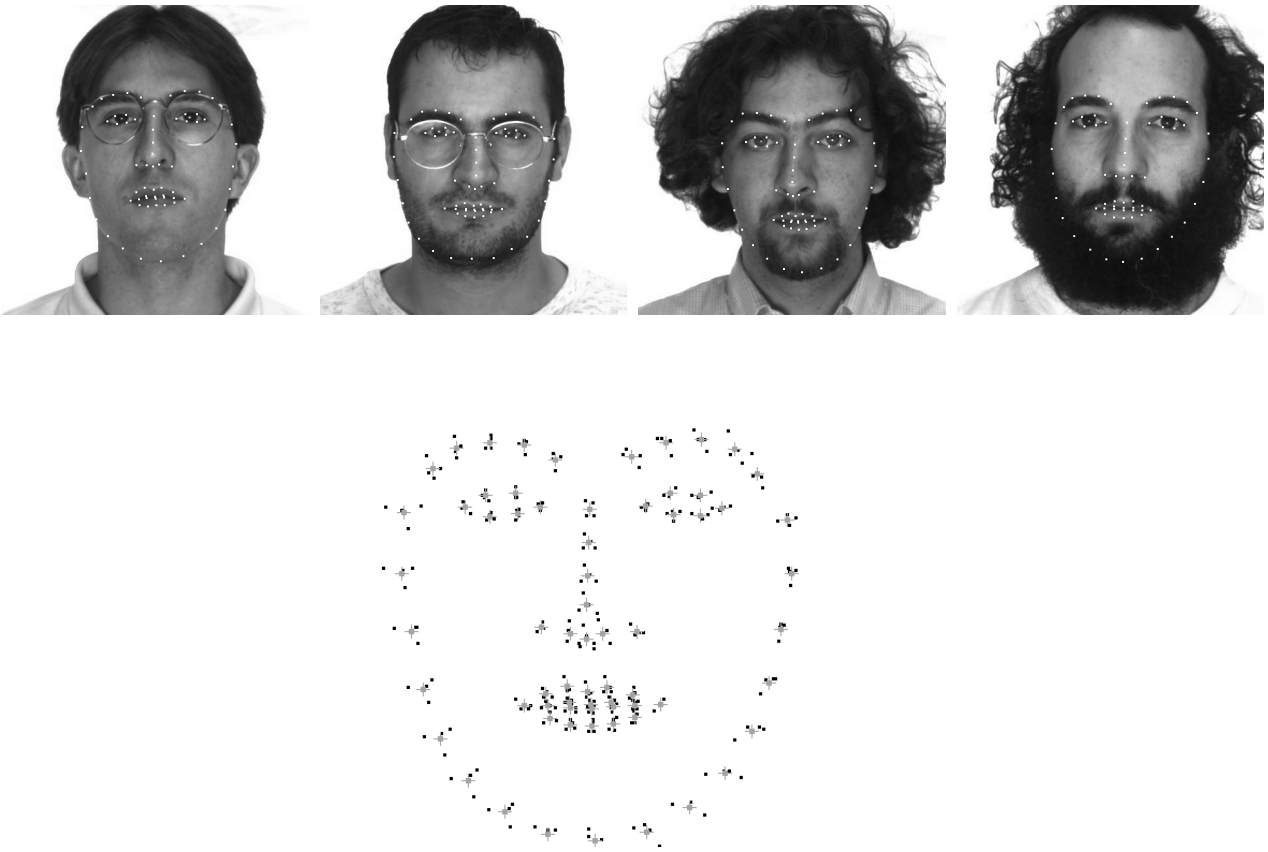


Figure 2.1: Four face configurations (representated on top) are brought to the shape space (bottom) where they are represented with black dots. The mean shape $\boldsymbol{s_0}$ is represented with grey crosses.

the sum of euclidean distance between corresponding landmarks on both shapes. Whereas aligning two 3D shapes implies to use an involved iterative optimization process (such algorithm can be found in [39]), the alignment of two 2D shapes has a closed form solution that allows to find a simple direct solution for the parameters of scale and orientation. This is useful since we will often be given to use this alignment algorithm. The algorithm is proposed in [34, Appendix B].

On steps (3) and (5), the scaling of the estimate mean shape brings it onto the surface of the unitary ray hypersphere of the nd dimensional space. On step (4), the alignment of all shapes onto the mean estimate bring those shapes to the surface of the hypersphere, or somewhere close to it. If the shape variation is large among the training data, a large non-linearity can be introduced because of the data distribution on the (non-linear) hypersphere surface. An approach to linearize the data consists in projecting the shapes onto the tangent space as done on step (6). The tangent space is the hyperplane of the nd dimensional space that is tangent to the unitary hypersphere and passes through $\boldsymbol{s_0}$. This step ensures the shape data to belong to a linear space, what simplifies the representation of the distribution by Principal Component Analysis as will be done in next paragraph.

Modelling shape variation. Once the shapes s_i are expressed in the same co-ordinate frame they are described by a distribution that we can analyse and represent. With this shape representation, we

expect to model the training data, as well as other similar, but distinct data. In particular, we seek a parameterised model of the form $\boldsymbol{s} = M(\boldsymbol{p})$, where $\boldsymbol{p}$ is a vector of parameters for the shape model. Such a model can be used to generate new vectors, $\boldsymbol{s}$.

An effective approach is to apply Principal Component Analysis (PCA) to the data. PCA is a statistical tool that transforms the space such that the covariance matrix is diagonal (i.e., it decorrelates the data). The data form a cloud of points in the vd-dimensional space. PCA computes the main axes of this cloud, allowing one to approximate any of the original points using a model with fewer than vd parameters. As well as dimensionality reduction, PCA also allows a linear representation of the data, what is desirable to design the fitting process described in this chapter in 2.2.1.

We now describe how the PCA is applied on shape data:

(1) Retrieve the mean shape $\boldsymbol{s}_0$ from the previous computation.

(2) Compute the covariance matrix of the data according to the following equation:

$$Cov = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{s}_i - \boldsymbol{s}_0)(\boldsymbol{s_i} - \boldsymbol{s}_0)^T \tag{2.2}$$

(3) Compute the eigenvectors $\boldsymbol{S_i}$ of Cov and the corresponding eigenvalues E_i. Sort eigenvectors so that E_i is higher than E_{i+1}.

In the case where we have N example vectors in a high dimensional space D, where $D > N$ finding the eigenvectors of Cov (which is $D \times D$) can be computationally expensive when D is large. When shape is represented with a sparse model as we use here, D is about 100 which is small enough for easily computing the eigenvectors on a 100×100 covariance matrix, but for appearance that we will deal with in 2.1.2, D is taken around one and ten thousand, which is too large to allow a classical eigenvector computation. Fortunately, we can equivalently solve a much simpler problem. Let $\boldsymbol{A} = [(\boldsymbol{s}_1 - \boldsymbol{s}_0)\ldots(\boldsymbol{s}_i - \boldsymbol{s}_0)\ldots(\boldsymbol{s}_N - \boldsymbol{s}_0)]$ be the $D \times N$ matrix of stacked observations where each column represent one observation. The unbiased covariance equation (2.2) becomes

$$Cov = \frac{1}{N-1}(\boldsymbol{A}\boldsymbol{A}^T) \tag{2.3}$$

where $(\boldsymbol{A}\boldsymbol{A}^T)$ is $D \times D$.

The solution consists in computing the eigenvectors and eigenvalues of the much smaller $N \times N$ $(\boldsymbol{A}^T\boldsymbol{A})$ matrix and transform the result. This can easily be seen by considering the definition of an eigenvector $\boldsymbol{x}$ of $(\boldsymbol{A}^T\boldsymbol{A})$ with eigenvalue λ:

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \tag{2.4}$$

we can then left-multiply both sides by $\boldsymbol{A}$ and regroup terms:

$$(\boldsymbol{A}\boldsymbol{A}^T)(\boldsymbol{A}\boldsymbol{x}) = \lambda(\boldsymbol{A}\boldsymbol{x}) \tag{2.5}$$

The vector $\boldsymbol{A}\boldsymbol{x}$ is thus an eigenvector of $\boldsymbol{A}\boldsymbol{A}^T$ with eigenvalue λ.

This only proves that the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ transformed by $\boldsymbol{A}$ are valid eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$. However, since $\boldsymbol{A}\boldsymbol{A}^T$ is $N \times N$, the question remains as to which N of its D eigenvectors we are getting (since $\boldsymbol{A}^T\boldsymbol{A}$ is only $N \times N$).

The answer is simple: since $\boldsymbol{A}$ only had N independent columns to begin with, the rank of $\boldsymbol{A}\boldsymbol{A}^T$ is only N. As a result, $\boldsymbol{A}\boldsymbol{A}^T$ only has N eigenvectors with non-zero eigenvalues. Note that $\boldsymbol{A}^T\boldsymbol{A}$

also is rank N and has N eigenvectors with the same non-zero eigenvalues as $\boldsymbol{A}\boldsymbol{A}^T$. Therefore, the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$ corresponding to non-zero eigenvalues can be found by transforming the eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$ by $\boldsymbol{A}$. Lastly, note that although the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$ and Cov are the same, we must scale the eigenvalues by $1/(N-1)$ to find the eigenvalues of Cov.

In reality, since the observations are centered onto the origin, the rank of Cov is not N, the number of data, but $N-1$ since we remove one degree of freedom by subtracting the mean data from all data to build the observations.

What we eventually obtain is a set of deformation vectors $\boldsymbol{s}_i$ that deform the mean shape $\boldsymbol{s}_0$ in a way that new shape instances $\boldsymbol{s}$ can be generated. Usually the n eigenvectors of higher eigenvalue are retained to build the generative model:

$$\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} p_i \boldsymbol{s}_i \tag{2.6}$$

where p_i are the shape coefficients weighting the deformation along the shape deformation vectors $\boldsymbol{s}_i$.

A constraint can be placed on the deformation coefficients p_i in such a way that the shape instance is plausible. It is usual to see that p_i are bounded to $\pm 3\sqrt{E_i}$, where $\sqrt{E_i}$ equals to the standard deviation of projected data onto the vector $\boldsymbol{s}_i$.

Among the consistant vectors available (those of non-zero eigenvalues) each vector represents a percentage of the global shape variance contained in the set of training examples. The percentage of shape variance attributed to $\boldsymbol{s}_i$ is given by the ratio

$$\frac{E_i}{\sum_{k=1}^{N-1} E_k} \tag{2.7}$$

Thus the percentage of total training variance represented by the first n shape vectors equals

$$\frac{\sum_{i=1}^{n} E_i}{\sum_{k=1}^{N-1} E_k} \tag{2.8}$$

Please note that PCA usually assumes that the data is gaussianly distributed. With face shape and appearance data, this is not always the case and PCA could be questioned but this is out of the scope of this thesis.

Let's now present the appearance component construction of an AAM.

2.1.2 The appearance component

To form a complete AAM one must consider shape as well as pixel intensity. To stress this point observe that shape is only well defined by inferring from knowledge of the pixel neighborhood. One must also consider the information beared by the pixels themselves.

In the following a scheme for capturing pixel information, using image warping, and modeling pixel variation, using principal component analysis, is described.

In the shape case, the example data already belong to the vd-dimensional space, where they can be normalized and then analyzed. In the texture case one needs a consistent method for collecting the texture information between the landmarks, and to project them into a co-ordinate frame that is common to all examples.

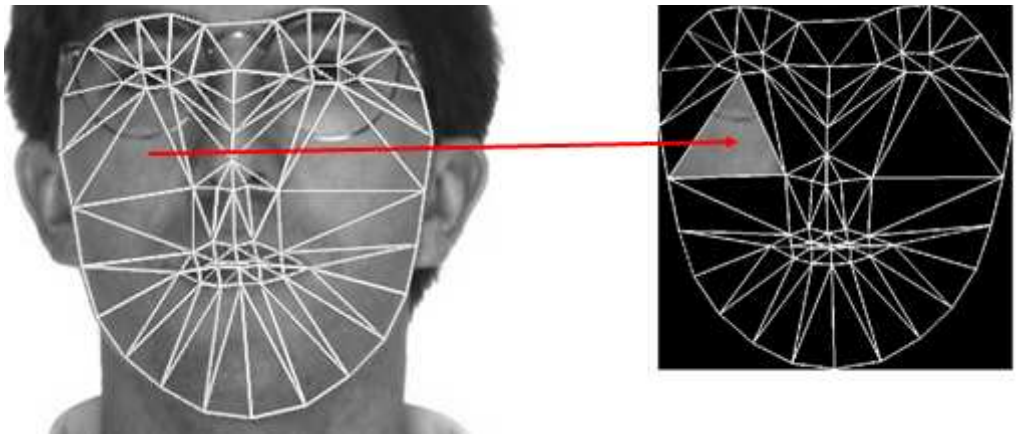


Figure 2.2: Warping of one image triangle into the corresponding $\boldsymbol{s}_0$'s triangle. Each pixel in the mean shape triangle picks up its intensity in the corresponding pixel in the image triangle.

Appearance representation. The mean shape $\boldsymbol{s}_0$, will be used as reference co-ordinate frame to express the appearance of each training example. We will now describe how this is performed.

A Delaunay triangulation is performed on $\boldsymbol{s}_0$'s set of landmarks, then obtaining a particular linkage indicating which landmark is connected to which others. Since the number of landmarks is equal between the mean shape and all training examples, the same triangulation is applied on all examples: the same linkage is used to establish the connections between landmarks, then forming a triangle mesh that can be placed into correspondence with $\boldsymbol{s}_0$'s.

Then a warping function must be established to warp the content of one training image's triangles into the $\boldsymbol{s}_0$'s corresponding triangles. On Figure 2.2 is shown the result of warping one image triangle into the corresponding mean shape's triangle. We make use of a simple affine warp.

The warp is performed by applying the triangular mesh of the first point set in the image to the second point set in $\boldsymbol{s}_0$. Let $\boldsymbol{x}^1$, $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$ denote the three vertices (landmarks) of the considered triangle in $\boldsymbol{s}_0$, and $\boldsymbol{x}_I^1$, $\boldsymbol{x}_I^2$ and $\boldsymbol{x}_I^3$ the corresponding vertices on the image triangulation. Then each point in the image triangle can be uniquely mapped upon the corresponding triangle of the reference frame $\boldsymbol{s}_0$ by an affine transformation. In practice what we do is to fill each pixel of the $\boldsymbol{s}_0$'s triangle with pixel intensity of the pixel that is in correspondence in the triangle on the image. Coordinates of each pixel $\boldsymbol{x}$ in $\boldsymbol{s}_0$'s triangle can be represented in function of the triangle vertices:

$$\boldsymbol{x} = \boldsymbol{x}^1 + \beta(\boldsymbol{x}^2 - \boldsymbol{x}^1) + \gamma(\boldsymbol{x}^3 - \boldsymbol{x}^1) \tag{2.9a}$$

$$= \alpha\boldsymbol{x}^1 + \beta\boldsymbol{x}^2 + \gamma\boldsymbol{x}^3 \tag{2.9b}$$

Thus $\alpha = 1 - (\beta + \gamma)$ giving $\alpha + \beta + \gamma = 1$. To constrain $\boldsymbol{x}$ to stay inside the triangle we must have $0 \leq \alpha, \beta, \gamma \leq 1$. The associated pixel $\boldsymbol{x}_I$ in the image is defined at the same coordinate β,γ in the coordinate system given by the vertices $\boldsymbol{x}_I^1$, $\boldsymbol{x}_I^2$ and $\boldsymbol{x}_I^3$

$$\boldsymbol{x}_I = \alpha\boldsymbol{x}_I^1 + \beta\boldsymbol{x}_I^2 + \gamma\boldsymbol{x}_I^3 \tag{2.10}$$

Given the three points of a triangle it is trivial to determine α,β and γ by solving the system of the two linear equations given by (2.9) for a known point, $\boldsymbol{x} = (x, y)^T$:

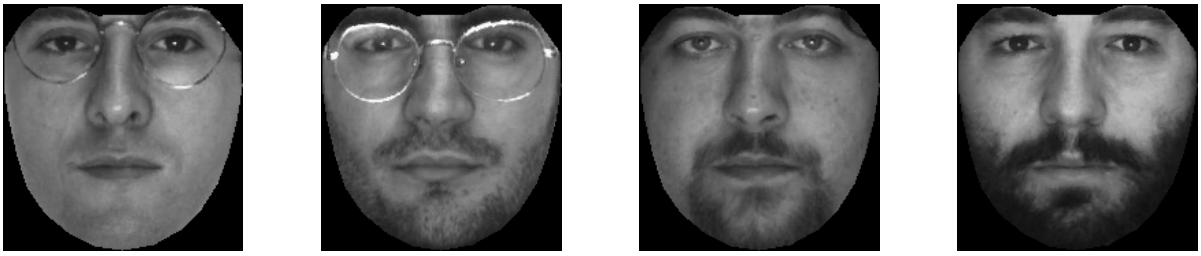


Figure 2.3: Appearances of the four faces seen on Figure 2.1 that have been mapped to the mean shape.

$$\alpha = 1 - (\beta + \gamma) \tag{2.11a}$$

$$\beta = \frac{(x - x^1)(y^3 - y^1) - (y - y^1)(x^3 - x^1)}{(x^2 - x^1)(y^3 - y^1) - (y^2 - y^1)(x^3 - x^1)} \tag{2.11b}$$

$$\gamma = \frac{(y - y^1)(x^2 - x^1) - (x - x^1)(y^2 - y^1)}{(x^2 - x^1)(y^3 - y^1) - (y^2 - y^1)(x^3 - x^1)} \tag{2.11c}$$

The intensity pixel of $\boldsymbol{x}$ is then given by the corresponding pixel $\boldsymbol{x}_I$ in the image that is defined by

$$\boldsymbol{x}_I = \boldsymbol{x}_I^1 + \beta(\boldsymbol{x}_I^2 - \boldsymbol{x}_I^1) + \gamma(\boldsymbol{x}_I^3 - \boldsymbol{x}_I^1) \tag{2.12}$$

Following the notation used in [75] we will note $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ the pixel $\boldsymbol{x}_I$ of the input image, which is considered to be the warped position of pixel $\boldsymbol{x}$ by the shape transformation $\boldsymbol{p}$.

Since the resulting coordinates will generally not produce positions on the integer pixel lattice of the image we simply grab the closest pixel to the one given by equation (2.12). More accurate pixel interpolation schemes, like Thin Plate Splines, could be used but would highly increase the computation time of the operation.

This operation is repeated for each pixel of each triangle on the reference frame $\boldsymbol{s}_0$ thus warping the entire reference frame onto the image and subsequently mapping the pixel values onto the reference frame. The process done on several images leads to collect the appearance training data necessary to build the appearance component of the global model. An example of such appearance collection is shown on Figure 2.3.

There are two main advantages in collecting the appearance data in this way. First the appearance data are all sampled into a common reference frame that is of defined and fixed size. The data can therefore be handled in a common framework and a statistical analysis can be performed on them. The second is that appearances are aligned and semantically equivalent face elements among training data (here faces) are placed into correspondence in the shape normalized reference frame. The analysis of global or local appearance variations makes sense since the variation applies to semantically corresponding elements/features.

In practice to speed up the process, the computation of β and γ is done once on all $\boldsymbol{s}_0$'s pixels at a pre-computation stage, and a lookup table is built, associating to each pixel of the mean shape $\boldsymbol{s}_0$ the reference of the triangle it belongs to, and the values of β and γ.

The appearance information of training data i is represented by a vector $\boldsymbol{g}_i$. This vector here contains the grey level pixel values but it could contain the RGB components of pixels found within the form of the object we are modeling. We will write:

$$\boldsymbol{g}_i = (g_{i1}, ..., g_{il}, ..., g_{iw})^T \tag{2.13}$$

where w is the number of pixels lying on the object surface. In this work g_{il} will represent the grey level intensity of the pixel l of data i.

Photometric normalization. As we previously filtered out the pose from the object to obtain the true shape, one would like the texture model to be invariant to global changes in illumination. Effects that cause such changes include usage of different film media, different exposure times, external lightning or shadows for instance. Below we will compensate for linear changes by applying a scaling α and an offset β. If $\boldsymbol{g}_i$ denotes the actual pixel values sampled in the image i:

$$\boldsymbol{g}_i \leftarrow \frac{\boldsymbol{g}_i - \beta\mathbf{1}}{\alpha} \tag{2.14}$$

where $\mathbf{1}$ is the unitary vector of length w: $\mathbf{1} = [1, 1, \ldots, 1]^T$.

The aim is to align each vector $\boldsymbol{g}_i$ onto the normalized mean vector $\overline{\boldsymbol{g}}$. The values of α and β are chosen to best match this normalized mean. If n is the number of examples, iterate the following three steps until the mean data $\overline{\boldsymbol{g}}$ is stable:

(1) compute an estimate of the mean appearance data:

$$\overline{\boldsymbol{g}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{g}_i$$

(2) normalize $\overline{\boldsymbol{g}}$ so as it is zero mean and of norm 1.

(3) use equation (2.14) to align each $\boldsymbol{g}_i$ onto the estimate mean shape: for each $\boldsymbol{g}_i$ compute the values

$$\alpha = \boldsymbol{g}_i^T.\overline{\boldsymbol{g}}, \quad \beta = (\boldsymbol{g}_i^T.\mathbf{1})/w \tag{2.15}$$

Modelling appearance variations The normalized appearance data are gathered and a PCA is applied on them following the same process described for shape. m eigencomponents of appearance associated to the highest eigenvalues are generally collected to represent the training data.

The generative model of appearance is given by the following linear expression:

$$\boldsymbol{A} = \boldsymbol{A}_0 + \sum_{i=1}^{n} p_i \boldsymbol{A}_i \tag{2.16}$$

Similarly to the shape deformation components $\boldsymbol{s}_i$, each component $\boldsymbol{A}_i$ of appearance variation represents a certain percentage of the total variance of appearance contained in the set of training examples. This percentage can be calculated on the basis of the eigenvalues associated to each eigencomponent, exactly as it was done for shape components.

The AAM is eventually the combination of the statistical shape generative model expressed by equation (2.6) and the statistical appearance generative model expressed by (2.16).

2.2 Image Alignment Algorithms

Image alignment typically consists to find the optimal warping parameters that bring a reference template image to make it resemble with good approximation to an area of an input image that contains the same object (up to a deformation that the warp is suppose to recover). Let's have a tour of the art of image alignment from the initial Lucas and Kanade's formulation to the Baker and Matthews formulation that we will use to fit an AAM on a picture.

2.2.1 The Lucas-Kanade Image Alignment Algorithm

Everything started with the introduction of the Lucas-Kanade algorithm used to compute optical flow in 1981 [70]. The template image, usually a small block of pixels, was grabbed into one frame in a video sequence, and translation on both axis was authorized to search for the same pattern in the subsequent frame. The maximum similarity between the template an the image was achieved through an optimization on the translation parameters. Gradient descent is a standard approach to perform this optimization. Many algorithms are available to perform the gradient step approximation, such as Gauss-Newton, Newton, steepest-descent, and Levenberg-Marquardt.

The basic method consists to minimize the image difference between a reference template $\boldsymbol{T}$, and a region of the input image $\boldsymbol{I}$. Two coordinate systems must be considered: the image coordinate system that describes with two coordinates (x_I, y_I) the location of the pixel $\boldsymbol{x}_I$ on $\boldsymbol{I}$, and the template coordinate system that describes with two coordinates (x, y) the location of the pixel $\boldsymbol{x}$ on $\boldsymbol{T}$. Correspondence between template and image coordinate systems is obtained with a parameterized warping operator $\boldsymbol{W}$. For a given vector $\boldsymbol{p}$ of warp parameters, $\boldsymbol{W}$ places each pixel $\boldsymbol{x}$ of $\boldsymbol{T}$ into correspondence with a pixel $\boldsymbol{x}_I$ of $\boldsymbol{I}$. We use the following notation:

$$\boldsymbol{x}_I = \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$$

When $\boldsymbol{p}$ varies, the corresponding pixel $\boldsymbol{x}_I$ to a given pixel $\boldsymbol{x}$ generally changes too.

The intensity level of pixel $\boldsymbol{x}$ of $\boldsymbol{T}$ can be noted $\boldsymbol{T}(\boldsymbol{x})$. This intensity pixel can be compared to the intensity level of pixel $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$ that we will note $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}))$.

The optimization task consists in minimizing the error between corresponding pixels in $\boldsymbol{T}$ and in $\boldsymbol{I}$, and this for all pixels of $\boldsymbol{T}$. For conveniency of representation and in accordance to the technical implementation of the algorithm we introduce an intermediary pixel coordinate. The patch $\boldsymbol{P}$ will receive the pixels warped from $\boldsymbol{I}$ through $\boldsymbol{W}$. $\boldsymbol{P}$ is of same dimension as the template $\boldsymbol{T}$ and we will also refer to $\boldsymbol{x}$ as a pixel of this template. The error must then be minimized between $\boldsymbol{P}$ and $\boldsymbol{T}$.

This minimization problem can be formulated in the following sum of squares manner:

$$\sum_{\boldsymbol{x}} [\boldsymbol{E}(\boldsymbol{x})]^2 \tag{2.17}$$

$$\sum_{\boldsymbol{x}} [\boldsymbol{P}(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.18}$$

$$\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{x}_I) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.19}$$

$$\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.20}$$

This can be solved as a least squares optimization task. When $\boldsymbol{p}$ are simply two translation parameters, it boils down to the original Lucas-Kanade optical-flow formulation where a small block of reference pixels was search in a frame by translation around a starting position. However a large

variety of warps of any complexity can be introduced and $\boldsymbol{p}$ can be a large set of parameters involving any combination (linear or not) between pixel $\boldsymbol{x}$'s coordinates (x, y).

Finding the best parameters $\boldsymbol{p}$ is a non-linear optimization problem. An iterative approach must be set up. Due to the complex and non-linear nature of images, the problem is also non-convex, and the method is generally prone to get stuck into local minimas. Lucas-Kanade algorithm assumes that a current estimate of $\boldsymbol{p}$ is known and then iteratively solves for increments to the parameters $\boldsymbol{\Delta p}$; i.e. the following expression is (approximately) minimized:

$$\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p} + \boldsymbol{\Delta p})) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.21}$$

A $\boldsymbol{\Delta p}$ estimation is computed at each iteration and the parameters $\boldsymbol{p}$ are additively updated:

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \boldsymbol{\Delta p} \tag{2.22}$$

The computation of parameter update $\boldsymbol{\Delta p}$ is obtained through a Gauss-Newton Gradient Descent approach.
The non-linear expression in equation (2.21) is linearized through a Taylor expansion applied on $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p} + \boldsymbol{\Delta p}))$ to give:

$$\sum_{\boldsymbol{x}} \left[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) + \nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - \boldsymbol{T}(\boldsymbol{x}) \right]^2 \tag{2.23}$$

where $\nabla \boldsymbol{I}$ represents the gradients of image $\boldsymbol{I}$, $\left(\frac{\partial \boldsymbol{I}}{\partial x}, \frac{\partial \boldsymbol{I}}{\partial y}\right)$, that have been warped back into the template coordinate frame with the current estimate of the warp $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$. The term $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is the Jacobian of the warp. If $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) = (W_x(\boldsymbol{x}, \boldsymbol{p}), W_y(\boldsymbol{x}, \boldsymbol{p}))^T$ then:

$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_n} \end{pmatrix} \tag{2.24}$$

The closed form solution of equation (2.23) is obtained by derivation of this equation with respect to $\Delta \boldsymbol{p}$ and set it equal to zero.

$$2 \sum_{\boldsymbol{x}} \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right]^T \left[\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) + \nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - \boldsymbol{T}(\boldsymbol{x}) \right] \tag{2.25}$$

where $\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is refered to as *Steepest Descent Images* (*SDIs*). The closed form solution of $\boldsymbol{\Delta p}$ in equation (2.25) is given by the following formula:

$$\Delta \boldsymbol{p} = H^{-1} \sum_{\boldsymbol{x}} \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right]^T [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})] \tag{2.26}$$

where H is the $n \times n$ Gauss-Newton approximation of the Hessian matrix

$$H = \sum_{\boldsymbol{x}} \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right]^T \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \right] \tag{2.27}$$

The Lucas-Kanade algorithm then consists in computing equations (2.26) and (2.22) iteratively until convergence is reached on parameter estimation. The major drawback of this optimization process is its elevated computational cost. Since gradients $\nabla \boldsymbol{I}$ must be evaluated at $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$ and Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$

The Lucas-Kanade Algorithm

Iterate:

(1)	Warp $\boldsymbol{I}$ in $\boldsymbol{P}$ $(= \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})))$ with $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})$	$[\mathrm{O}(nw)]$
(2)	Compute the error image $\boldsymbol{E}(\boldsymbol{x}) = [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})]$	$[\mathrm{O}(w)]$
(3)	Warp the gradient $\nabla \boldsymbol{I}$ with $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})$	$[\mathrm{O}(nw)]$
(4)	Evaluate the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ at $(\boldsymbol{x},\boldsymbol{p})$	$[\mathrm{O}(nw)]$
(5)	Compute the steepest descent images $\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$	$[\mathrm{O}(nw)]$
(6)	Compute the Hessian matrix using equation (2.27)	$[\mathrm{O}(n^2 w)]$
(7)	Compute $\sum_x \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \boldsymbol{E}(\boldsymbol{x})$	$[\mathrm{O}(nw)]$
(8)	Compute $\Delta \boldsymbol{p}$ using equation (2.26)	$[\mathrm{O}(n^3)]$
(9)	Update the parameters $\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta \boldsymbol{p}$	$[\mathrm{O}(n)]$
	Total Computational Cost per iteration	$[\mathrm{O}(n^2 w + n^3)]$

Figure 2.4: Steps involved in the computation of one iteration of the Lucas-Kanade algorithm. The computational cost (in number of operations) of each step is reported on the right side, with n the number of shape components, and w the number of pixels that define the reference template. .

at $\boldsymbol{p}$, they both depend on $\boldsymbol{p}$ and must generally be recomputed at each iteration. In the particular case where the warp is the translation on both x and y axis, the jacobian does not depend on $\boldsymbol{p}$ and is then constant, what speeds up the computation. However for general warps all gradients and jacobian must be computed at each iteration since $\boldsymbol{p}$ varies from iteration to iteration.
The algorithm is summerized on Figure 2.4.

The computation cost of the total algorithm per iteration is $[\mathrm{O}(n^2 w + n^3)]$, where n is the number of parameters used for warping and w is the number of pixels that compose the template $\boldsymbol{T}$. The Lucas-Kanade algorithm has the advantage to present very few restrictions: the only requirement to compute the jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is that the warp $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})$ is differentiable with respect to the warp parameters $\boldsymbol{p}$. The main drawback of this algorithm is that it is computationally expensive and in general does not allow real time applications.

Since [70], some researchers have investigated the possibility to minimize differently the cost function: $\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})]^2$. They proposed some interesting approaches to the problem, all provably equivalent to the Lucas-Kanade's one (both empirically and theoretically to first order approximation in the parameter update $\Delta \boldsymbol{p}$). Besides the interest ensuing from the originality of these approaches, they present the technical advantage to modify the algorithm and bring substantial reduction of the overall computational cost.

Shum and Szeliski [98] proposed an *additive compositional* approach where the parameter update $\Delta \boldsymbol{p}$ is not directly estimated on the image coordinate frame but in the current warped image $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}))$ instead. In the Lucas-Kanade approach, the optimization problem could be formulated by the following question: how to warp the input image in order to make it more resemble to the reference template $\boldsymbol{T}$? In the Shum and Szeliski's additive compositional framework, their approach stands in the question: how to warp the current warp to make it more resemble to the template? How to warp $\boldsymbol{P}$ to better match $\boldsymbol{T}$? The equation to minimize then becomes

$$\sum_{\boldsymbol{x}} [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{W}(\boldsymbol{x},\Delta \boldsymbol{p}),\boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})]^2 \tag{2.28}$$

Where the warp $\boldsymbol{W}(\boldsymbol{x},\Delta \boldsymbol{p})$ is estimated at each iteration and consequently composed to the current warp as follows

$$\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x},\Delta \boldsymbol{p}) \tag{2.29}$$

This approach has the advantage that the Jacobian matrix does not depend on $\boldsymbol{p}$ anymore and can therefore be pre-computed and left constant accross the iterations. The compositional approach requires the set of used warps to form a *semi-group*, *i.e.* , it must contain the identity warp and they must be closed under composition. This condition is respected for many warps, including homographies and 3D rotations, but usually does not stand for piecewise affine warps (like AAMs) that do not form a group. See [9] for more details on the additive compositional framework.

Hager and Belhumeur [58] proposed an *inverse additive* approach, claiming that the key to efficiency is switching the role of the image and the template. Their approach can be presented by the following question: how can we warp the reference template to make it more resemble to the input image? A change of variable gives $\boldsymbol{y} = \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$ and then $\boldsymbol{x} = \boldsymbol{W}(\boldsymbol{y}, \boldsymbol{p})^{-1}$, what involves changes in the original cost function that becomes

$$\sum_{\boldsymbol{y}} \left| \frac{\partial \boldsymbol{W}^{-1}}{\partial \boldsymbol{y}} \right| . \left[\boldsymbol{I}(\boldsymbol{y}) - \boldsymbol{T}(\boldsymbol{W}(\boldsymbol{y}, \boldsymbol{p})^{-1}) \right]^2 \tag{2.30}$$

For a restricted set of warps, the Jacobian matrix can be factored and the part that depends on the parameters can be treated separately from the iterative process. It results in an efficient algorithm but the restriction on possible warps is strong: mainly 2D similarity transform, 2D affine warps, and a small number of more esotoric warps are allowed.

Baker and Matthews [8] recently proposed the *inverse compositional* algorithm. Since the implementation we use for fitting AAM is based on this approach, it will be the object of next section.

In comparison to these approaches, the Lucas-Kanade's is also called *forward additive* approach for it computes the parameter update in a forward manner and it applies it in an additive way.

Please note that a complete review of all image alignment approaches has been thoroughly done in [9] as well as in [22]. In the latter work, Buenaposada also proposed further investigation on the inverse compositional algorithm that gave positive exit for 3D plane movement registration. In [11] can also be found the general answer regarding the question of 2.5D and pure 3D alignment through inverse composition.

2.2.2 The Inverse Compositional Image Alignment Algorithm

Baker and Matthews extended the investigation on image alignment methods and proposed in [8] the original inverse compositional algorithm in 2001. Their approach can be sumarized by the following question: how can we warp the reference template $\boldsymbol{T}$ in order to make it more resemble to the image warp $\boldsymbol{P}$? The warp estimate must be inverted to come back to the forward-compositional problem of making $\boldsymbol{P}$ more resemble to $\boldsymbol{T}$. Then the update can be composed to the current warp on image $\boldsymbol{I}$. The equation to minimize is proposed to be the following

$$\sum_{\boldsymbol{x}} \left[\boldsymbol{T}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) \right]^2 \tag{2.31}$$

Where the warp $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})$ is estimated at each iteration and consequently inverted and composed to the current warp as follows

$$\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})^{-1} \tag{2.32}$$

The Taylor expansion of equation (2.31) gives

$$\sum_{\boldsymbol{x}} \left[\boldsymbol{T}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{0})) + \nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \Delta \boldsymbol{p} - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) \right]^2 \tag{2.33}$$

The Original Inverse Compositional Algorithm

Pre-Compute:

(3)	Evaluate the gradient $\nabla \boldsymbol{T}$ of template $\boldsymbol{T}$	[O(w)]
(4)	Evaluate the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ at $(\boldsymbol{x}, \boldsymbol{0})$	[O(nw)]
(5)	Compute the steepest descent images $\nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$	[O(nw)]
(6)	Compute the Hessian matrix $\boldsymbol{H}$ using (2.35) and invert it	[O(n^2w)]
	Total Computational Cost at pre-computation	[O(n^2w)]

Iterate:

(1)	Warp $\boldsymbol{I}$ in $\boldsymbol{P}$ $(= \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})))$ with $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$	[O(nw)]
(2)	Compute the error image $E(\boldsymbol{x}) = [\boldsymbol{T}(\boldsymbol{x}) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}))]$	[O(w)]
(7)	Compute $\sum_x \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T E(\boldsymbol{x})$	[O(nw)]
(8)	Compute $\Delta \boldsymbol{p}$ using equation (2.34)	[O(n^3)]
(9)	Update the parameters $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x}, \Delta \boldsymbol{p})^{-1}$	[O(n^2)]
	Total Computational Cost per iteration	[O($nw + n^3$)]

Figure 2.5: Precomputation steps and iteration steps of the inverse compositional algorithm.

It is assumed that $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{0}))$ is the identity warp. The derivation of equation (2.33) gives the closed form solution for the update $\Delta \boldsymbol{p}$

$$\Delta \boldsymbol{p} = H^{-1} \sum_{\boldsymbol{x}} \left[\nabla \boldsymbol{I} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T [\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) - \boldsymbol{T}(\boldsymbol{x})] \tag{2.34}$$

where the Hessian matrix H varies from equation (2.35) for that $\boldsymbol{I}$ is replaced by $\boldsymbol{T}$

$$H = \sum_{\boldsymbol{x}} \left[\nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right]^T \left[\nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right] \tag{2.35}$$

and the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is evaluated at $(\boldsymbol{x}, \boldsymbol{0})$. Nothing in the Hessian depends on $\boldsymbol{p}$: it is constant across iterations and can therefore be pre-computed. Gradients, Jacobian matrix and Hessian can all be sent to the pre-computation stage, what results in a very efficient algorithm. Figure 2.5 sumarizes all computation steps of both pre-computation stage and online iterative stage.

This approach presents the same requirement than the forward compositional approach: the set of warps must form a semi-group. Besides this condition, it is necessary that the update warp $\boldsymbol{W}(\boldsymbol{x}, \Delta \boldsymbol{p})$ can be inverted before it is composed with the current estimate. The set of warps must then form a group. Most of warps do form a group, including homographies and 3D rotations. Unfortunately this is not the case for AAMs and for other piecewise affine warps. Approximations must be made to extend the inverse compositional algorithm to these warps. In [8], Baker and Matthews show how they operate this extension. We will present this in 2.3.

In [9], Baker and Matthews have empirically illustrated the difference of timing per iteration between the various solutions proposed. This illustration was given on an example implemented in Matlab where they use a $\boldsymbol{T}$ is a 100×100 pixels template image and a 6-parameter affine warp. The Forward Compositional showed to be 1.5 times faster than the Lucas-Kanade algorithm, the Inverse Additive 13.4 times faster, and the Inverse Compositional 13.8 times faster than the original Lucas-Kanade.

2.3 AAM fitting - some implementation details

Baker and Matthews developed the Inverse Compositional (IC) framework to use it for AAM fitting as a main application. Fitting an AAM onto an image consists in performing the alignment of the reference template on the image. The correct alignment of the template onto the image should bring the model vertices into their corresponding position on the object to analyse on the image. We deal here with faces, then the alignment of the face template onto the input image should bring the model vertices onto their corresponding position on the face represented on the image: the semantical meaning of these positions can be used for high level post-treatment, namely here face and facial deformation analysis. We will refer to the alignment process as *fitting*. We will say that the model is fitted onto the image. Sequence tracking is typically obtained by fitting the model on one image, and to hand-over the fitting solution to the subsequent frame as the initial position for the fitting onto the new image. The face motion is considered to be small, so that the optimization process only requires few iterations to fit one frame of the sequence, and the model can avoid to be stuck into local minima as it can happen when the model is initialized too far from the optimal position.

Already existing methods were not completely satisfactory in the sense that they had to resort to non-gradient descent to obtain efficiency, or they make some easy assumptions to leave constant the gradients in an additive framework [28], what is fondamentally incorrect and limits AAM fitting performances. The correctness of the mathematical formulation of the image alignment problem is important for maximizing the system performance. In the other hand, it is true that a forward additive approach like the Lucas-Kanade's one is particularly slow and is not suitable for real-time applications on the current standard material.

The IC algorithm opened up the possibilities to fit AAMs efficiently. However as we mentioned before, the problem comes from the incorrectness of the IC algorithm to deal with warps like AAMs that do not form a group.

First approximation is made on the inversion of the AAM update $\Delta\boldsymbol{p}$. It is simply taken as $-\Delta\boldsymbol{p}$, which is an approximation since $\boldsymbol{W}(\boldsymbol{x}, \Delta\boldsymbol{p})) \circ \boldsymbol{W}(\boldsymbol{x}, -\Delta\boldsymbol{p}))$ does not generally results in the identity warp. However this approximation is small. Second approximation comes from the composition of the inverted parameter update to the current warp. The warp composition makes losing the connexity between the adjacent triangles that compose the model: for one single vertex of the model, the shape update (the inverted one) that must be composed to the current warped position of this vertex on the image is defined in the $\boldsymbol{s}_0$ frame as a displacement quantity that can be expressed in one coordinate frame per triangle adjacent to the vertex. The composition is applied on the current warped vertex on the image by applying to this vertex the same displacement quantities, calculated in the adjacent triangle coordinate frames, what usually brings the vertex into many different locations since the triangles usually have on the image a different geometry from the one they have in the $\boldsymbol{s}_0$ frame. Please see §3.2.3 of [76] for more details.

The solution to this problem is somewhat tricky and consists to consider the possible locations for one vertex and to take their average as the effective location. The approximation can be dangerous when the current warp particularly deforms the model triangles on the image, and when one becomes so flat that the coordinate frame it generates gets singular.

Once accepted these small approximations it is possible to operate AAM fitting under the efficient IC framework. In [8] Baker and Matthews shows the improved accuracy and speed of their method with respect to the one presented in [27]. In [54] Gay-Bellile *et al.* register paper-like surfaces with the use of Thin Plate Splines. With features directly driven in the reference template frame (and not resulting from the application of a set of shape parameters), they save a reprojection steps usually aiming at retrieving these shape parameters. Whereas the warp does not form a group, the inversion and composition are operated without any approximation.

We will briefly present the different steps of implementation of the algorithm for AAM fitting purpose. All complementary information can be found in [75]. Our approach, also the one used in [75], will be to first present the basic implementation where the template is constant and only AAM shape deformation is performed, then the four components that deal with the 2D similarity transform (two components for translation, one for scale and the last for rotation) will be added to the already present shape deformation components, and eventually explain how to allow appearance variations on the reference template in order to cope with appearance changes occuring on the input images.

2.3.1 Basic implementation

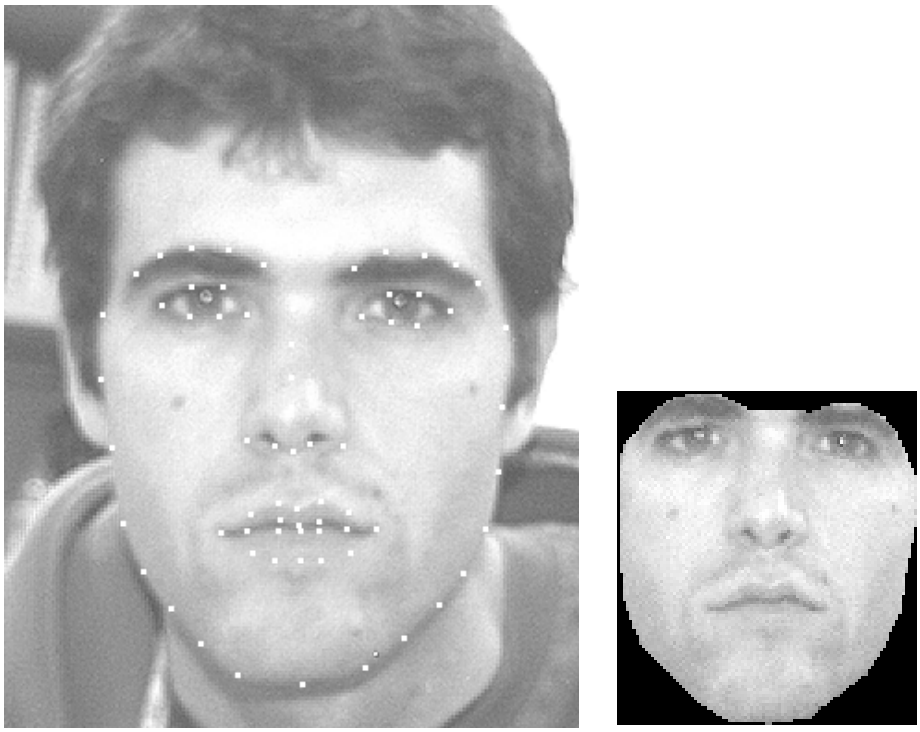
AAM is generally allowed to deform in shape, to be subject to 2D similarity transformations and to vary in appearance. The first step of the implementation addresses the problem of shape deformation.

The aim of this section will be to present the physical meaning of the various elements that play a role in the optimization process. We will also show how operations should be applied on these elements.

The template $\boldsymbol{T}$. The template $\boldsymbol{T}$ constitutes the reference face. It is the pattern that will be searched for in the input image. $\boldsymbol{T}$ can be represented as an image. Ideally, the visual information it contains should also be present in the input image, up to a shape deformation of the mean shape $\boldsymbol{s}_0$. This image is indeed represented in a patch of pixels that has the dimension of $\boldsymbol{s}_0$. Obviously to form an image, $\boldsymbol{s}_0$ must be scaled and translated in positive coordinates to be expressed in a pixel coordinate system, according to what we presented previously. The scaling parameter applied on $\boldsymbol{s}_0$ to form $\boldsymbol{T}$ defines the size in pixels of the template. Let's consider an example to foster the understanding. On Figure 2.6 on the right, we built a template $\boldsymbol{T}$ of $w = 8376$ pixels size. The reference face image that fills the template is taken from a training image visible on the left side of the same figure. The image appearance that lies inside the mesh defined by the manual labelling is warped onto the template as was explained in a previous part of the chapter.

As said before, we adopted a simple pixel picking method for warping the image into $\boldsymbol{T}$. More advanced methods could be used to optimize this warping operation and to avoid aliasing for example. This could slightly improve the fitting performances. Also the template size w has an influence on fitting accuracy (besides the fact that the fitting process speed highly depends on it). We observed that a too small template results in a loss of fitting accuracy: the process do not have enough information on the image to fit onto, and maybe some aliasing effect due to uncorrect downsampling operation can be encountered (the original image is downsampled when it is warped into a smaller destination one, and origin image should therefore be down-filtered before downsampling). We do not know what is the optimal size for the template. However we can observe that when the template size exceeds the size of the area of interest in the image(s) it models, improvements on fitting accuracy are not significant. A deeper investigation of these two aspects of warping method and template size is out of sake in this work. In an interesting work ([113]) concerned with video surveillance applications where people are located at very different distances from the camera, the authors experimentally show that the template size that maximizes the AAM performances should be taken slightly larger than the size of the face on the image. The size of training images is also chosen accordingly to the template size. They rely on several models to span a whole range of face sizes they must deal with.

The template gradient $\nabla\boldsymbol{T} = \left(\frac{\partial \boldsymbol{T}}{\partial x}, \frac{\partial \boldsymbol{T}}{\partial y}\right)$. The gradient images along both vertical and horizontal axes are required in the algorithm. There are different ways to compute the gradient of an image. We



Training image Template $\boldsymbol{T}$

Figure 2.6: Illustration of template $\boldsymbol{T}$ formation. The labelled training image on the left is warped onto the template frame on the right.

used a simple one where

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

and

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

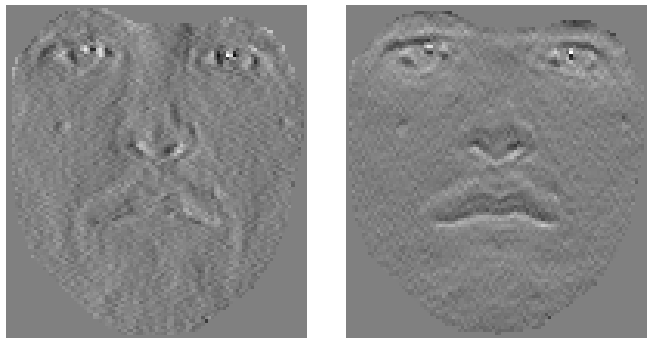
Pixels located on the template border are left to zero in the gradient images. Example of $\frac{\partial \boldsymbol{T}}{\partial x}$ and $\frac{\partial \boldsymbol{T}}{\partial y}$ is represented on Figure 2.7

It will be interesting to understand how much influence the method to compute the gradients has on the fitting performances. Once again, this understanding is out of sake here.

The Jacobian matrix $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$. The destination of the pixel $\boldsymbol{x}$ under the piecewise affine warp $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})$ depends on the AAM shape parameters $\boldsymbol{p}$ through the v vertices of the mesh $\boldsymbol{s}$ on the image. Recall that $\boldsymbol{s} = (x_1, \ldots, x_v, y_1, \ldots, y_v)^T$. Applying the chain rule to the warp $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})$ gives:

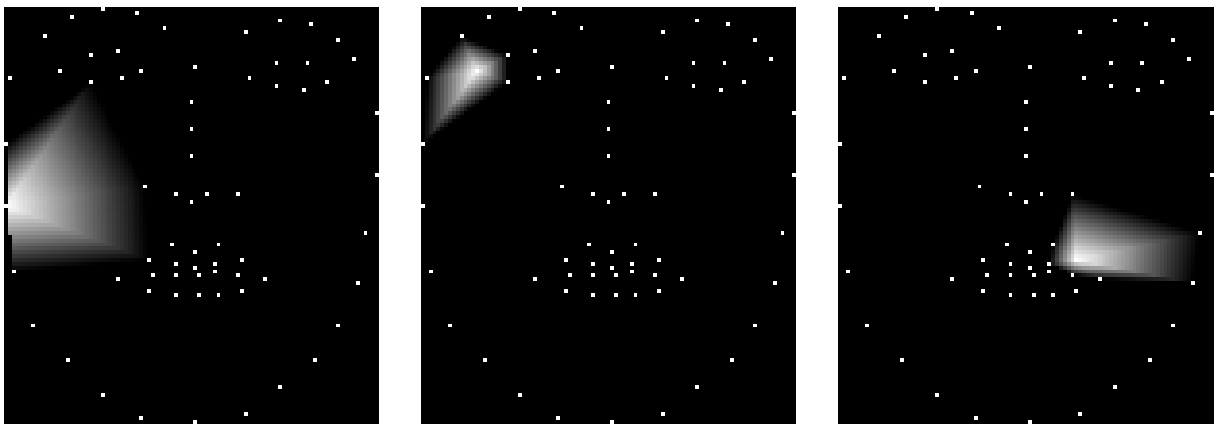
$$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} = \sum_{i}^{v} \left[\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x_i}} \frac{\partial \boldsymbol{x_i}}{\partial \boldsymbol{p}} + \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{y_i}} \frac{\partial \boldsymbol{y_i}}{\partial \boldsymbol{p}} \right] \tag{2.36}$$

Details on the mathematical meaning of $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x_i}}$ and $\frac{\partial \boldsymbol{x_i}}{\partial \boldsymbol{p}}$ can be found in [75] section 4.1.2. Here, we will simply give the practical way we built this Jacobian matrix.



$\frac{\partial \boldsymbol{T}}{\partial x}$ $\frac{\partial \boldsymbol{T}}{\partial y}$

Figure 2.7: Illustration of the template gradients along x and along y.



$\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_3}_x$ or $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial y_3}_y$ $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_{28}}_x$ or $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial y_{28}}_y$ $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_{54}}_x$ or $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial y_{54}}_y$

Figure 2.8: Illustration 3 images.

We recall from equation (2.12) that $\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}) = \boldsymbol{x}_1 + \beta(\boldsymbol{x}_2 - \boldsymbol{x}_1) + \gamma(\boldsymbol{x}_3 - \boldsymbol{x}_1)$. It results that

$$\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_i} = (1-\beta-\gamma, 0)^T, \quad \frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial y_i} = (0, 1-\beta-\gamma)^T \tag{2.37a}$$

$\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_i}$ is a vector of two images of size $\boldsymbol{T}$, the second receives zero value for each pixel and the first also receives zero value everywhere, except on the triangles that are linked to vertex i. Pixels lying in this triangle receives the value $1-\beta-\gamma$ where β and γ depends on the position of the pixel $\boldsymbol{x}$ considered in the triangles of vertex i's neighborhood. $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial y_i}$ is the same vector but both images are flipped. On Figure 2.8 we give some examples of the image that should be obtained for some vertices.

The second components of the Jacobian are $\frac{\partial \boldsymbol{x_i}}{\partial \boldsymbol{p}}$ and $\frac{\partial \boldsymbol{y_i}}{\partial \boldsymbol{p}}$. Differentiating equation $\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} p_i \boldsymbol{s}_i$, we obtain:

$$\frac{\partial \boldsymbol{x_i}}{\partial \boldsymbol{p}} = (\boldsymbol{s}_1^{x_i}, \boldsymbol{s}_2^{x_i}, \cdots, \boldsymbol{s}_n^{x_i}), \quad \frac{\partial \boldsymbol{y_i}}{\partial \boldsymbol{p}} = (\boldsymbol{s}_1^{y_i}, \boldsymbol{s}_2^{y_i}, \cdots, \boldsymbol{s}_n^{y_i}) \tag{2.38a}$$

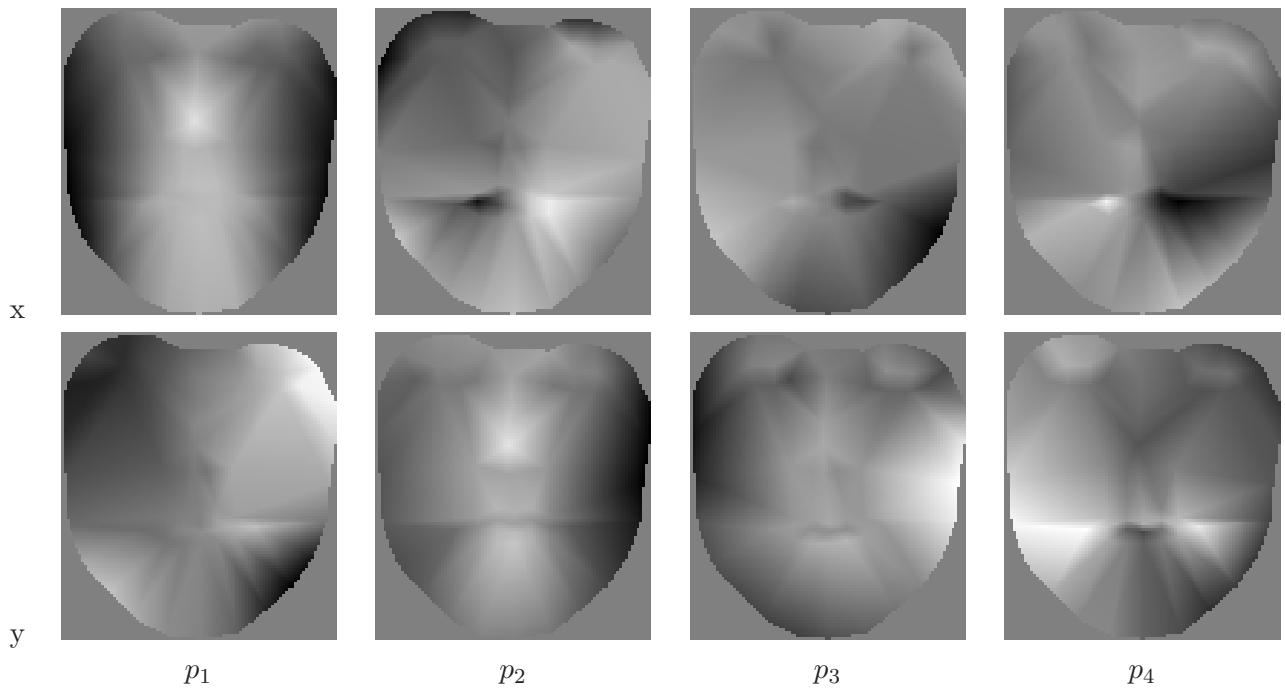


Figure 2.9: The first 2×4 elements of the Jacobian matrix.

where $\boldsymbol{s}_j^{x_i}$ is the element of shape component $\boldsymbol{s}_j$ that corresponds to x_i, and similarly for y_i. $\frac{\partial \boldsymbol{x_i}}{\partial \boldsymbol{p}}$ and $\frac{\partial \boldsymbol{y_i}}{\partial \boldsymbol{p}}$ are then simple rearangement of the shape components $\boldsymbol{s}_i$: they are vectors of scalars.

To compute equation (2.9) we need to compute $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{p}} + \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{y}_i}\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{p}}$ for each vertex i. $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{p}}$ is a $2 \times n$ matrix of images of size $\boldsymbol{T}$, the bottom ones are all composed of zero pixel values, the top ones are images $\frac{\partial \boldsymbol{W}(\boldsymbol{x},\boldsymbol{p})}{\partial x_i}_x$ where each is weighted with the ith element of $\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{p}}$. Similarly, $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{y}_i}\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{p}}$ is a $2 \times n$ matrix of images of size $\boldsymbol{T}$ where the top ones are all composed of zero pixel values, and the bottom ones are images $\frac{\partial \boldsymbol{W}(\boldsymbol{y},\boldsymbol{p})}{\partial y_i}_y$ where each is weighted with the ith element of $\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{p}}$.

$\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{x}_i}\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{p}} + \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{y}_i}\frac{\partial \boldsymbol{y}_i}{\partial \boldsymbol{p}}$ is eventually a $2 \times n$ matrix of images of size $\boldsymbol{T}$, that describe the motion tendency of vertex i for each deformation component. If we compute all $2 \times n$ matrices over all vertices, we can sum them over their corresponding elements: corresponding images are summed up to form the final $2 \times n$ Jacobian matrix $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$, also composed of images of size of $\boldsymbol{T}$. on Figure 2.9 is represented a Jacobian matrix for 4 modes of deformation. Deformation tendency along x and along y are described for each mode.

The Steepest Descent Images or SDIs, $\nabla \boldsymbol{T}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ This is actually a vector of n images. As describes it the formula $\nabla \boldsymbol{T}\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$, this vector is obtained by multiplication of the 1×2 template gradients matrix $\nabla \boldsymbol{T}$ by the $2 \times n$ Jacobian matrix. Multiplication of two images forms a new image: an element-wise product is performed between the elements that have the same coordinates on the two images, and the result constitutes the steepest descent image. n steepest descent images are then computed.

The Hessian matrix, $\boldsymbol{H}$ Finally the Hessian matrix, is computed by multiplication and sum of the SDIs as indicated in equation (2.35). Multiplication of the column vector $\left(\nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right)^T$ by the row vector $\left(\nabla \boldsymbol{T} \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\right)$ gives a $n \times n$ matrix of images. Each image is then summed over its elements to provide a resulting scalar and form the final Hessian matrix composed of $n \times n$ scalars.

Mind that $\boldsymbol{H}$ is symetric and only the diagonal elements and all elements relative to one side of the diagonal should be computed. All other elements from the other side of the diagonal can simply be copied from their symetric.

2.3.2 Dealing with shape similarities

Shape deformation of the mean shape s_0 is insufficient to perform a fitting on an input image. Global shape transformation, corresponding to 2D similarities (rotation, translation and scale of the model) is also required. Global shape deformation can be modeled by four vectors that are here refered to as $\boldsymbol{s}_i^*$. To refer to all the shape deformation parameters, we stack the vector $\boldsymbol{p}^*$ corresponding to the four global shape deformation parameters and the vector $\boldsymbol{p}$ corresponding to the internal shape deformation parameters. Details on the four extra components are left to the reader in [75]. For the explanation of the way the shape parameters can be retrieves, we abusively refer to $\boldsymbol{p}$ as the concatenated shape parameters $\boldsymbol{p}$ and $\boldsymbol{p}^*$.

Four more components increase the dimension of the Jacobian matrix (from $2 \times n$ to $2 \times (4+n)$), SDIs (from $1 \times n$ to $1 \times (4+n)$) and Hessian matrix (from $n \times n$ to $(4+n) \times (4+n)$).

Inversion and composition of the estimate warp They develop first order approximations to the inverse of a warp and the composition of two warps. These approximations are correct to the first order.

As we said before, the update is computed for the template $\boldsymbol{T}$ in the inverse compositional framework. To be applied to the model shape on the input image, the update must be inverted, and then composed to the current warp. The inversion and composition both require to make a first order approximation. Once done, the current warp is updated, what results in a displacement of the model vertices on the image with respect to the previous iteration. From this new position, the pixel warp $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}))$ can be performed without explicit knowledge of $\boldsymbol{p}$ since all vertices positions on the image are known, and this is enough to perform the triangle to triangle warp mapping.

It can however be interesting to retrieve the corresponding shape parameters $\boldsymbol{p}$, maybe if we want to apply some constraints in the parameter excursion. In [75], the authors propose to first retrieve the parameters $\boldsymbol{p}^*$, computing the quantities necessary to center, normalize and align on $\boldsymbol{s}_0$ the current model configuration of vertices. Once applied this similarity transformation to the configuration, it can thus be projected onto the orthonormal shape basis made of the $\boldsymbol{s}_i$ components. Parameters $\boldsymbol{p}$ are consequently retrieved.

We do a simple all-in-one retrieval step. We offline stack the shape components $\boldsymbol{s}$ and $\boldsymbol{s}_i^*$ to form the shape basis $\boldsymbol{B}_s = [s_1, \cdots, s_i, \cdots, s_1^*, \cdots, s_4^*]$.

Given a set of parameters $\boldsymbol{p}$, the corresponding shape $\boldsymbol{s}$ is obtained as follows: $\boldsymbol{s} = \boldsymbol{B}_s \boldsymbol{p}$.

For the reversed problem where we are given a configuration $\boldsymbol{s}$, since $\boldsymbol{B}_s$ is not orthonormal, we make use of the pseudo-inverse to retrieve the corresponding set of parameters $\boldsymbol{p}$.

$$\boldsymbol{p} = \tilde{\boldsymbol{B}}_s \boldsymbol{s} \tag{2.39}$$

where $\tilde{\boldsymbol{B}}_s = (\boldsymbol{B}_s^T \boldsymbol{B}_s)^{-1} \boldsymbol{B}_s^T$, and can be pre-computed offline.

2.3.3 Dealing with appearance variations

In [75], an extension was proposed in order to include appearance variations in the AAM fitting process. The linear appearance varying model proposed by equation (2.16) is included in the function to minimize. The fixed template $\boldsymbol{T}$ is now replaced by the appearance varying template $\boldsymbol{A}_0 + \sum \lambda_i \boldsymbol{A}_i$. The inconvenient of including appearance variations is that the steepest descent images depend on them, and SDIs, Jacobian and Hessian should be updated at each iteration, what spoils the efficiency of the algorithtm.

The Project-Out Algorithm. The authors propose a framework they call *project-out* where the shape parameters $\boldsymbol{p}$ are optimized leaving the appearance parameters $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m]$ to zero, quantities for which the gradients have been pre-computed offline. The appearance is however included in the minimization process through a projection of the appearance lying under the current model position onto the appearance basis.

This approximation has proved to maintain the efficiency of the algorithm, but it does not allow to deal with large appearance variations. Indeed, in [57] Gross et al. show that this approximation does not allow good fitting results in the case of unseen faces (faces that do not belong to the training set). The more correct formulation called *simultaneous* where shape and appearance parameters are optimized simultaneously allow better fitting results in the case of unseen faces with respect to the project-out version of the algorithm (see [57]).

The *Simultaneous Inverse Compositional* (SIC) algorithm presented in [4] is almost as slow as a forward-additive algorithm, but we decided to lead our investigations under this mathematically correct formulation in order to understand how best can perform the AAM in terms of accuracy and robustness in various contexts.

The Simultaneous Inverse Compositional Algorithm. The basic formulation is usually found written in the following manner:

$$\sum_{\boldsymbol{x}} \left[\boldsymbol{A}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(\boldsymbol{x}) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) \right]^2 \tag{2.40}$$

The parameter optimization consists in iteratively minimizing the following equation:

$$\sum_{\boldsymbol{x}} \left[\boldsymbol{A}_0(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})) + \sum_{i=1}^{m} (\lambda_i + \boldsymbol{\Delta}\lambda_i) \boldsymbol{A}_i(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})) \right]^2 \tag{2.41}$$

simultaneously with respect to $\boldsymbol{\Delta p}$ and $\boldsymbol{\Delta\lambda} = (\boldsymbol{\Delta}\lambda_1, \cdots, \boldsymbol{\Delta}\lambda_m)^T$.

We skip the derivative steps that can be found in part 3.1 of [4].

The formula for the steepest descent image becomes:

$$\boldsymbol{SD}_{sic} = \left(\boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_1}, \cdots, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_n}, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_1^*}, \cdots, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_4^*}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m \right) \tag{2.42}$$

where $\boldsymbol{\nabla A} = (\nabla \boldsymbol{A}_0 + \sum_{i=1}^{m} \lambda_i \nabla \boldsymbol{A}_i)$, where we see clearly the dependance to the current parameters $\boldsymbol{\lambda}$.

The error image is here expressed as:

$$\boldsymbol{E}_{sic} = \boldsymbol{A}_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(\boldsymbol{x}) - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})). \tag{2.43}$$

The Simultaneous Inverse Compositional Algorithm

Pre-Compute:

(3)	Evaluate the gradient $\nabla \boldsymbol{A}_i$ for $i = 1, \cdots, m+2$	$[\mathrm{O}((m+2)w)]$
(4)	Evaluate the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ at $(\boldsymbol{x}, \boldsymbol{0})$	$[\mathrm{O}((n+4)w)]$
	Total Computational Cost at pre-computation	$[\mathrm{O}((n+m+6)w)]$

Iterate:

(1)	Warp $\boldsymbol{I}$ with $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p})$	$[\mathrm{O}(w)]$
(2)	Compute the error image $\boldsymbol{E}_{sic} = \boldsymbol{B}_a \boldsymbol{\lambda} - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}))$	$[\mathrm{O}((m+2)w)]$
(5)	Compute the steepest descent images $\boldsymbol{SD}_{sic}$ using equation (2.47)	$[\mathrm{O}((n+m+6)w)]$
(6)	Compute the Hessian matrix $\boldsymbol{H}_{sic}$ using (2.45) and invert it	
(7)	Compute $\sum_{\boldsymbol{x}} \boldsymbol{SD}_{sic}^T(\boldsymbol{x}) \boldsymbol{E}_{sic}(\boldsymbol{x})$	$[\mathrm{O}((n+m+6)w)]$
(8)	Compute $\boldsymbol{\Delta p}$ using equation (2.44)	$[\mathrm{O}((n+m+6)^2)]$
(9)	Update the parameters $\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x}, \boldsymbol{\Delta p})^{-1}$ and $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \boldsymbol{\Delta \lambda}$	$[\mathrm{O}((n+4)^2 + (m+2))]$

Cost of (6) $= [\mathrm{O}(w(n+m+6)^2 + (n+m+6)^3)]$

Total Computational Cost per iteration $=$

$[\mathrm{O}(w(3m + 2n + 15 + (n+m+6)^2) + (n+m+6)^3 + (n+m+6)^2 + (n+4)^2 + (m+2)^2)]$

Figure 2.10: Precomputation steps and iteration steps of the simultaneous inverse compositional algorithm.

We have the new update expressed as:

$$(\boldsymbol{\Delta p}^T, \boldsymbol{\Delta \lambda}^T)^T = -H_{sic}^{-1} \sum_{\boldsymbol{x}} \boldsymbol{SD}_{sic}^T(\boldsymbol{x}) \boldsymbol{E}_{sic}(\boldsymbol{x}) \tag{2.44}$$

with

$$H_{sic} = \sum_{\boldsymbol{x}} \boldsymbol{SD}_{sic}^T(\boldsymbol{x}) \boldsymbol{SD}_{sic}(\boldsymbol{x}) \tag{2.45}$$

The update is applied in inverse and composition to the shape warp $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x}; \boldsymbol{\Delta v})^{-1}$, and in additive way for the appearance $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \boldsymbol{\Delta \lambda}$.

We also include two extra appearance components representing the offset level $\boldsymbol{A}_I$ (a w long vector of 1) and a gain $\boldsymbol{A}_0$ (the mean appearance patch itself). They are also used in [4]. They introduce some robustness to global light variations and photometric changes. All appearance components can now be gathered in a base $\boldsymbol{B}_a = [\boldsymbol{A}_1, \cdots, \boldsymbol{A}_m, \boldsymbol{A}_0, \boldsymbol{A}_I]$, and the cost function reformulated as:

$$\sum_{\boldsymbol{x}} [\boldsymbol{B}_a \boldsymbol{\lambda} - \boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x}, \boldsymbol{p}))]^2 \tag{2.46}$$

The only one change happens for the steepest descent images:

$$\boldsymbol{SD}_{sic} = \left(\boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_1}, \cdots, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_n}, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_1^*}, \cdots, \boldsymbol{\nabla A} \frac{\partial \boldsymbol{W}}{\partial p_4^*}, \boldsymbol{A}_1, \cdots, \boldsymbol{A}_m, \boldsymbol{A}_0, \boldsymbol{A}_I \right) \tag{2.47}$$

where now $\boldsymbol{\nabla A} = \sum_{i=1}^{m+2} \lambda_i \nabla \boldsymbol{A}_i$

Note that for $m = 0$ (only the gain and offset components are considered), Bartoli showed in [18] how the original efficiency of the Inverse Compositional framework can be conserved.

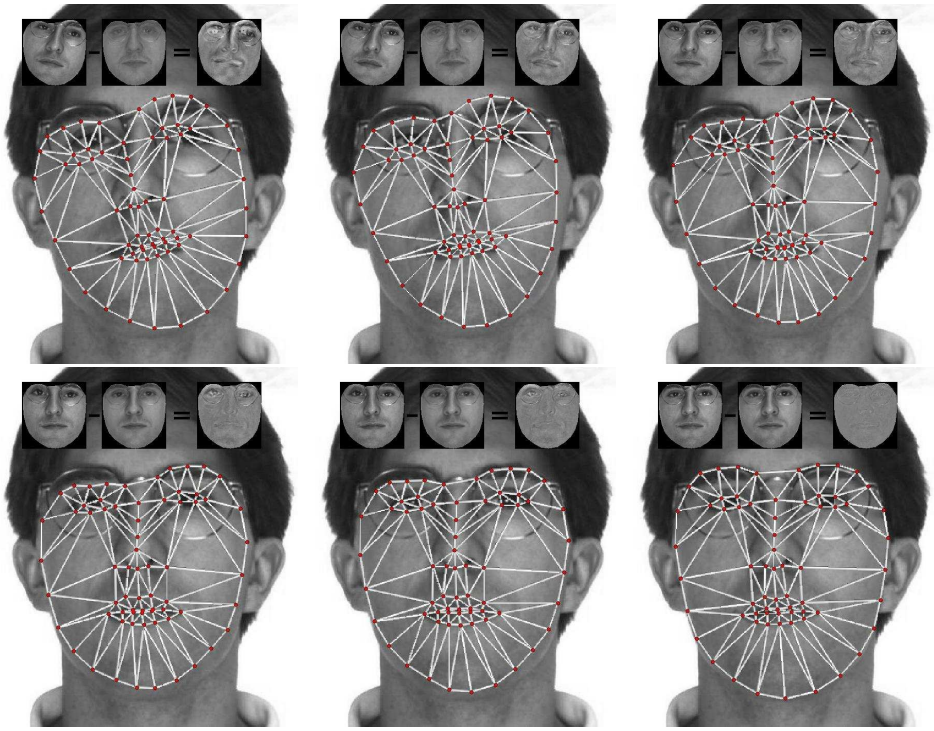


Figure 2.11: Example of the fitting process with the SIC. From top left to right bottom, iterations 1, 6, 10, 15, 18 and 28 are shown.

2.3.4 Illustration of the fitting process

Figure 2.11 gives an illustration of the fitting process obtained with the inverse compositional simultaneous algorithm. The model position is shown for six iterations of the process. On each image we also see (on the front of the person's face) three smaller images at size $\boldsymbol{s}_0$. The first represents $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}))$ the reprojection of the image pixels lying under the model onto the template of size $\boldsymbol{s}_0$ (previously called $\boldsymbol{P}$ in equation (2.18)). The second is $\boldsymbol{B}_a\boldsymbol{\lambda}$, the current appearance instance of size $\boldsymbol{s}_0$ too (previously called $\boldsymbol{T}$ in equation (2.18)). Finally the third is $\boldsymbol{E}_{sic}$, the error image. To minimize the square of the residual over the error image, the algorithm optimize the model position on the image and the model instance of appearance to maximize the correspondence between $\boldsymbol{I}(\boldsymbol{W}(\boldsymbol{x},\boldsymbol{p}))$ and $\boldsymbol{B}_a\boldsymbol{\lambda}$, as is done iteratively on this example.

Chapter 3

The Person-Specific Context

A look at the literature concerning the Active Appearance Models can easily leads the reader to think that with a large training set containing many identities, poses and expressions, the resulting AAM can generalize to any identity, pose and expression. I initially did think that myself. Indeed it might be a very good motivation to implement this method for a researcher who must deal with face and facial expression analysis. However, it is often not clear in the literature which images are used for training, and how many identities, poses and expressions are represented: a serious lack of details on experimental conditions makes difficult the evaluation a priori of the real performances of AAMs. AAM, probably like any method available for image analysis, presents several advantages but is not the holly graal method and it also suffers from many drawbacks.

Maybe the really first contribution of this thesis is to clearly reveal to the reader the true advantages and drawbacks of AAMs. When do AAMs perform well, and when they do not? In 3.1, we will present the main contexts of use of an AAM, showing for each context the way the AAM behaves, either finely or poorly. The two following parts will focus on the context in which the AAM performs best: the *person-specific context*, where the AAM fits on the data that are used for training it. A first behavior observation will be done in 3.2 before deeper characterizing in 3.3 the performances on an AAM in this context.

Let's now have a look at different contexts of use to have a clearer understanding of the way an AAM can perform.

3.1 The various contexts in which to use an AAM

Hereafter we present the main context of use of an AAM. The *person-specific* context is shown under various forms. This is the qualification attributed in [57, 75] to an AAM that specifically learnt the face pictures is must fit onto. We show the context of illumination variation, and eventually, we show the *generic* context. Borrowing [57]'s terminology, an AAM is said to be *generic* when it is able to fit a face not explicitely learnt.

Person-specific context: first case study. A first look at the works presented in the literature hardly present the difficulties that are met when using AAMs. The method can even seem magic to the reader since results are often obtained under the best context one can use an AAM which is the person-specific context. In this context, the AAM is trained on the base of images that are explicitely extracted from the same video sequence that is analysed, and moreover, training images are chosen to represent the most significant events of the sequence. This is not often said that clearly in literature. Let's present the results we can obtain by training an AAM on 19 keyframes thoroughly

extracted to globally express the information contained in a 350 frame video sequence. 95% of shape and appearance total variance are retained as it is usually the case in most proposed works.

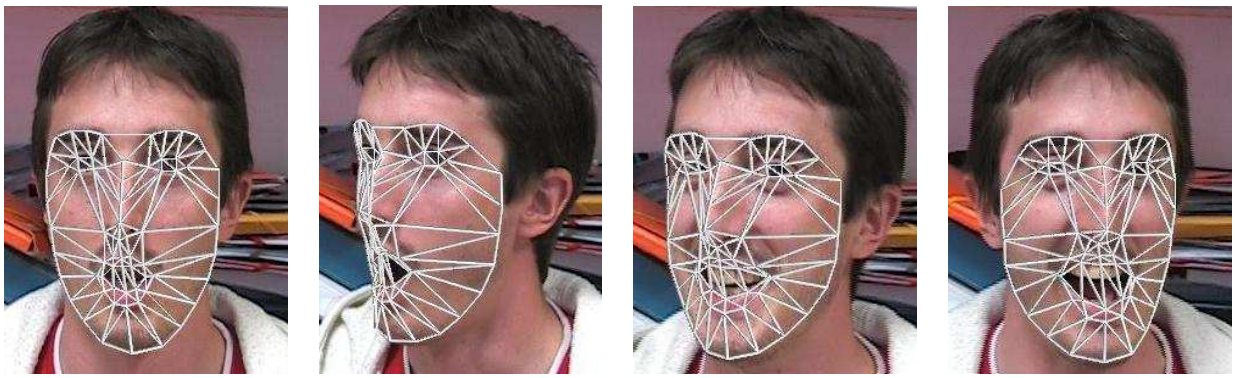


Figure 3.1: Frames number 121, 218, 252 and 274 of the 350 frame tracked sequence. In this *person-specific* context, where keyframes from the tracked sequence are used to train the AAM, the fitting result is impressively accurate.

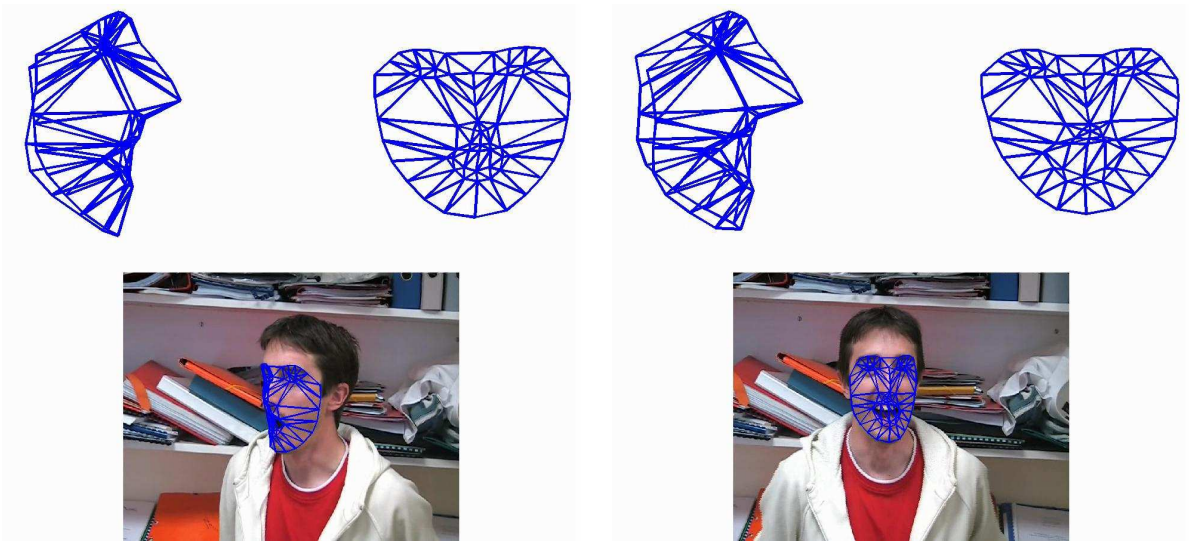


Figure 3.2: The 2D tracking is accurate enough to allow 3D reconstruction via (non-rigid) Structure From Motion, as an example of the possible post treatments before properly dealing with analysis and interpretations. For two frames of the tracked sequence (frames 218 and 274), the face configuration is reconstructed in 3D and displayed under two views, frontal and lateral.

The tracking result is simply impressive as is shown on figure 3.1. Even the defects manually introduced while labelling the training key-frames are smoothed and the model perfectly sticks to every movements and deformations. The vertices position at each frame can reliably be used to perform any kind of post-analysis. In [71] for instance, Simon Lucey *et al.* work in this context to retrieve AUs from spontaneous facial expressions. In [104], Torresani *et al.* recently proposed a non-rigid Structure From Motion algorithm allowing to calibrate the camera and retrieve the 3D mean shape structure of the face as well as a set of 3D deformable components. A 3D reconstruction using

a similar fashion was used by Gay-Bellile *et al.* to reconstruct the sequence. An illustration of 3D reconstruction of the face shape in the tracked sequence is shown on figure 3.2.

However, this context is rather limitative and only allows to operate some offline processings at the price of a tedious manual labelling of tens of frames for each sequence.

Illumination changes. In the person-specific context (as well as in a generic context), the AAM is unrobust to illumination changes it has not explicitely learnt.

Figure 3.3 convincingly illustrates this fact.

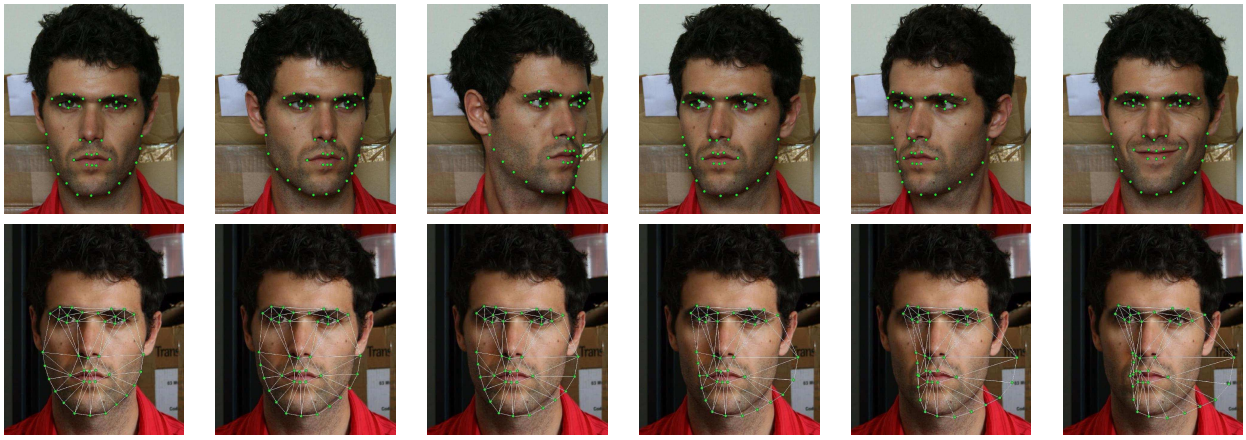


Figure 3.3: On top row, 6 frames used to train an AAM are represented. The face is homogeneously illuminated on these frames. The AAM is fitted onto a picture, identical to the first training frame, except for the illumination which here is lateral. The iterative process is shown on this picture on the bottom row at initialization (given manually), and for iterations 2, 4, 12, 24 and 30. The fitting process diverges and shows to be clearly unrobust to illumination changes.

In chapter 6 we originally and successfully address this problem.

One can now wonder what happens if we deal with the same face and same expressions and also same illumination used for training the AAM, but on a different sequence?

Person-specific context: second case. A variant of the person-specific context for an AAM consists to learn a face under a given illumination and for a range of poses and expressions, and to fit it onto the same face on a different sequence under the same conditions. Figure 3.4 illustrates a result obtained with an AAM trained on 72 pictures (the two next section will make use of these training pictures that are well described in section 3.2) and fitted onto a similar but distinct sequence.

Tracking results are still very interesting. They are very accurate whenever the pose and expression displayed are explicitely learnt. When the expression is new to the training set, it is usually partially or badly recovered by the AAM. Indeed, every single facial expression displayed for the tracked poses should be learnt explicitely to be fitted accurately. Also, the global model used to model the whole face should learn all combinations of upper and lower facial expressions: the AAM cannot fit onto some lower and upper expressions learnt separately.

Let's now increase the difficulty and observe what happens when keyframes from different people under different expressions are learnt.

Person-specific context: third case. In [75], Matthews and Baker presented a comparison test where one person is tracked with an AAM that has learnt only images from the sequence to be tracked

Frame 27 Frame 79 Frame 116 Frame 133 Frame 206 Frame 294 Frame 314

Figure 3.4: For this tracking illustration the AAM is trained on data representing the same face present on the sequence, but these data are not extracted from the sequence itself. When the face displays configurations known to the AAM, the fit is accurate. When the configuration is not explicitely learnt, the AAM lacks of accuracy. The fifth frame illustrates this phenomena on the eyebrows poorly fitted by the AAM. Any new data we wish to fit accurately should be learnt explicitely.

on the one hand, and where the same sequence is tracked with an AAM that has learnt the same images, plus images taken from four more different people sequences in the second hand. Tracking robustness decreases in the second case, where the tracked sequence is learnt among others: the tracker loses in robustness when it loses its specificity. In context of fitting on *known data* (where the AAM has learnt what it must fit onto) the AAM performances decrease when increases the quantity of training data.

We repeated such a test to understand the phenomena. We here briefly illustrate it on figure 3.5. We selected two frames from the test sequence. The second frame is 7 frames farther from the first one in the sequence, and the head consistently moved during this interval of time. On the first frame the AAM is initialized very well on the face.

From this initial position, we run an AAM specialized only on the tracked person's face, and another AAM that learnt this face, plus other 4 people's. We observe their ability to fit the second frame correctly. The two fitting processes are illustrated on figure 3.5. The first one, where the AAM is lighter, shows to reach the correct fitting solution in less iterations than the second AAM that, since having a more complex shape and appearance space, needs more iterations to reach the solution, and also shows a higher lack of robustness. Indeed, the model gets *in extremis* out of a local minima into which it was about to get stuck.

Tracking robustness decreases in the second case, where the tracked sequence is learnt among others: the tracker loses in performance when it loses its specificity. In context of fitting on *known data* the AAM robustness decreases when increases the quantity of learnt data.

Generic context. One could expect the AAM to perform well in the most generic case where unseen identity varies with pose and expression. If we stuff a large amount of identities, poses and expressions into one single AAM, it will be likely to considerably lack of robustness. Moreover, any data that is known only partially could not be fitted accurately, but this will be further investigated in chapter 4.

On figure 3.6 we illustrate a tracking result in the generic case where the AAM is trained on the 57 pictures displaying the 4 people different from the tracked person. The person is thus excluded from the training set.

The tracking is possible, but it lacks of accuracy and robustness what makes impossible any accurate post analysis.

A generic test with more training data could be done to observe how much the accuracy can be improved, and how much fitting speed and robustness should be sacrificed in exchange.

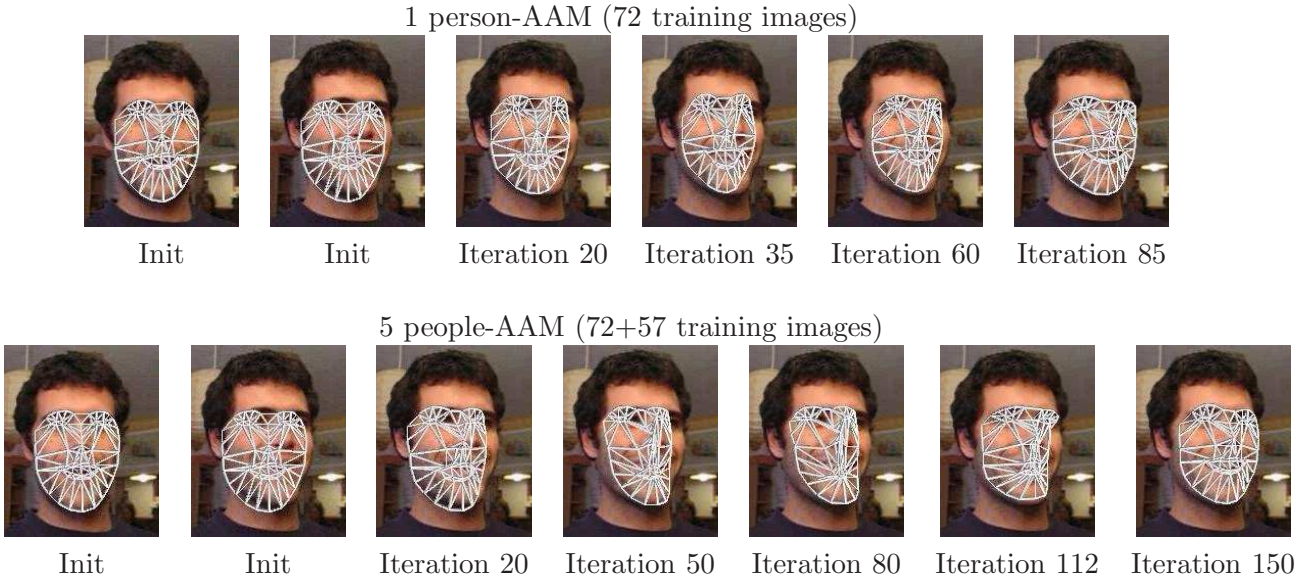


Figure 3.5: First and second rows respectively illustrate the fitting process for the AAM specialized on this person only, and for the AAM training on this person plus on 4 other people. The fitting process shows more robustness and speed to reach the correct solution.

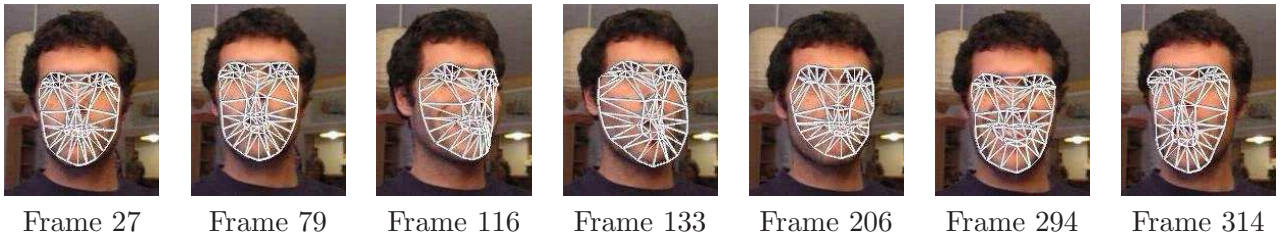


Figure 3.6: An AAM is trained with 57 pictures representing 4 people. The tracked person is excluded from the training data. In this generic tracking context, the AAM presents an important lack of fitting robustness and accuracy. Any post analysis based on such a fitting result would be unreliable.

Discussion. These preliminary experiments helped to have a first understanding of the way an AAM performs.

The generic context works bad, and this is very disappointing because we expected to be able to deal with unseen faces with changing expression and varying pose. Many applications require this ability. These applications might thus be out of reach for the AAMs.

Despite the apparent difficulty to use an AAM in a generic context, and despite the fact that it was one of our basical requirement to be able to deal with this context, we decided not to throw away the AAM solution, but instead to explore it deeply. After all, can't we manage to respulse the current limitations to make some steps further and overcome the specificity to one person or to one illumination? While the rest of this chapter focuses on the person-specific context, the following ones will aim to better explore the possibility to extend AAM performances to fit unseen faces, and under varying illumination.

We observed through these first experiments that an AAM can perform very well in the person-specific context where what must be fitted is also what is learnt by the AAM. We can thus rely on this

context to build person-specific applications. But as soon as we increase the amount of learnt data by adding extra people pictures, the AAM becomes more complex and starts to lose fitting robustness and speed.

Since highest fitting performances are required, and performances on known data decrease when increases the amount of learnt data, the best way to use an AAM in the known context is when the minimum amount of useful information is learnt. Then an application in the known context should focus on the single person, and the specific poses and expressions it must be able to retrieve. The less information we must learn, the better performances will be obtained.

But in the one-person case, can we keep adding training pose and expression data of this person and keep constant the fitting performances, or are we also subject to a performance loss when data increase in terms of learnt poses and expressions? This is the question we will try to answer in the following.

3.2 The person-specific context: first observations

As we explained, we want to know more about the ability of the AAM to learn and track one unique person's pose and expression variations. This ability would be useful for person-specific applications where, at the price of an initial fastidious manual labelling of tens of images, the system would work for one person, tracking him/her accurately and also allowing to extract his/her expressions under pose variations.

We show that the fitting/tracking can be evaluated according to different criteria which are the robustness (to initialization perturbation), the accuracy, and the processing speed. The parameters having an influence on fitting robustness and fitting accuracy are the quantity (and nature) of data in the training set and the amount of variance retained for shape and for appearance. In their tests, Baker *et al.* [75] as well as Cootes [28] usually retain 95% or 98% variance of shape and appearance: these values are typically used by the AAM users in the community. This unjustified choice for using the fitting method under such a fixed parameterization make their tests uncomplete.

We mean here to have a better understanding of AAM fitting behavior on known data accross various parameterizations: we will study the effect of quantity of learnt data and amount of shape and appearance variance retained.

For this test, we built a set of five video sequences representing the same subject displaying the same serie of eight facial expressions representing the following typical emotions: neutral, smile (invisible teeth), open smile (visible teeth), surprise, anger, disgust and fear. Each sequence is acquired at a different angle varying horizontally. Frontal, 20°, 30°, 35°, 45° sequences have been acquired. The lack of synchronized video cameras required to perform five distinct acquisitions, which is not a crucial problem for our test.

From each sequence we extracted eight key-frames where the appex of each expression is displayed. We labeled each image manually following the 68 point label fashion of [75]. To obtain the symetrical complementary pictures, we flip the training images relative to 20°, 30°, 35° and 45° sequences, together with their training points. Four complementary sequences are built in this way: −20°, −30°, −35° and −45° sequences. We make the assumption that the face is symetrical. However we do not mean to study this difference in performance here. Figure 3.7 illustrates available training images relative to 45°, 30°, 0°, −30° and −45° sequences.

For this test, we build two AAMs with different amount of training images. We will build the first AAM on the eight training images relative to the frontal sequence, and will call it $AAM_{frontal}$. The second AAM will be built on the seventy-two training images relative to the nine sequences available and we will call this second model $AAM_{AllPoses}$.

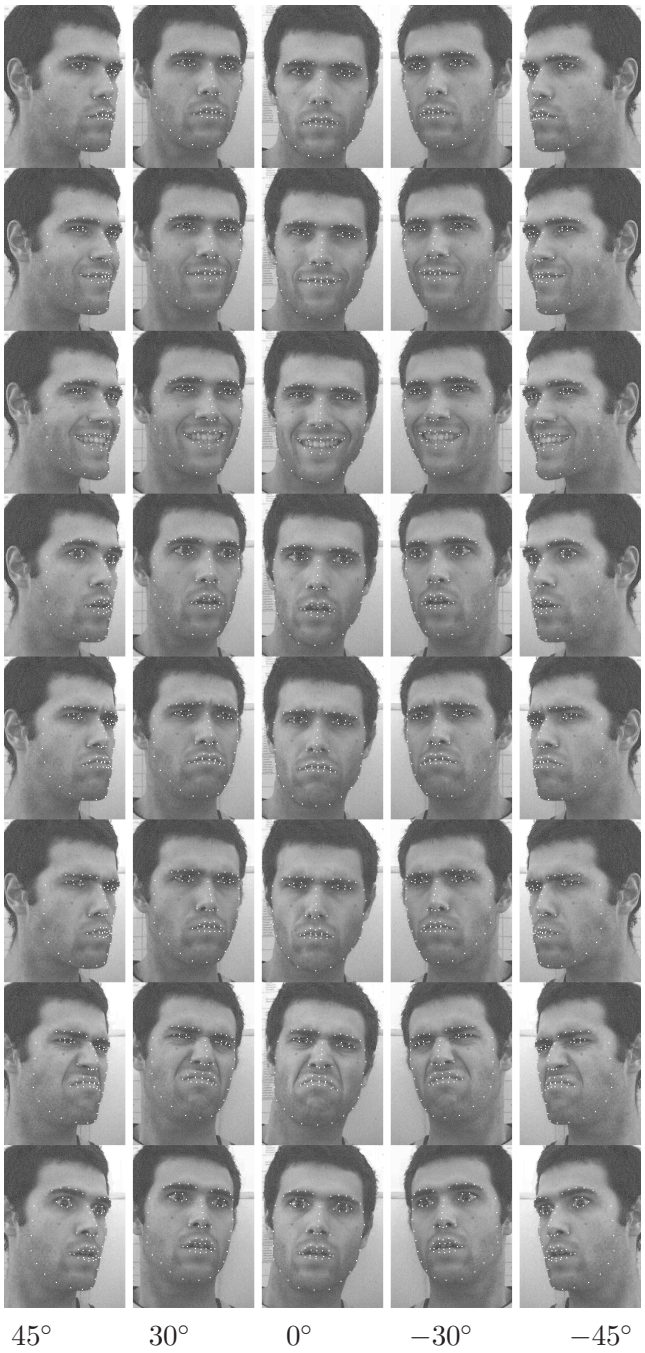


Figure 3.7: Extract of the 72 images used for training the $AAM_{AllPoses}$. Only the 8 pictures relative to the frontal sequence are used to train the $AAM_{frontal}$.

In this part we will track the frontal sequence with both AAMs for varying amounts of shape and appearance variance retained. This test will give us a first idea of the AAM behavior with respect to the three parameters shape variance, appearance variance, and amount of data in the training set. We will observe the tracking behavior and determine three performance criterion that should be analysed further: fitting accuracy at convergence, robustness to perturbations from the global minima position,

and speed of convergence.

In part 3.3 we will try to quantify both AAMs characteristics with respect to these criterion. For this purpose, we will operate a complete test to check AAMs performances according to their possible parameterization in shape and appearance percentage of variance retained from the training set, and starting distance from the ideal position.

Since both AAMs *know* the frontal sequence, we want to compare their behavior on this sequence.

The tracking is performed as follows: we launch the model in the optimal position on first frame of the sequence, the model is iteratively fit on the picture and last iteration gives the fitting result for this frame, and finally we hand-over the fitting result as initial position for fitting on the subsequent frame. The procedure is repeated for all frames of the sequence. We choose to fix the number of iterations per frame to 20. Other stopping criteria could be used but we observe that they are not always reliable. We determined this number of iterations in order to ensure that convergence is reached whenever it can be. As few as 4 or 5 iterations are usually necessary to converge. If we previously set the process to more iterations per frame, it was to ensure the convergence. No more than six iterations are usually satisfactory when the frame rate is high, or the head movement are not too fast.

For the $AAM_{frontal}$, as will be done for the $AAM_{AllPoses}$ too, it is interesting to observe the fitting behavior for different amounts of shape and appearance variance. We will not perform an exhaustive test, but only scan various shape variance for an appearance variance fixed to 95%, and then scan various appearance variance for a shape variance fixed to 95%. With 95% variance, the model is able to reconstruct enough information to allow the observation of the other varying parameter effect on fitting. We do not expect to learn anything new by scanning exhaustively all possible combinations of shape and appearance variance.

3.2.1 Tracking the frontal sequence with $AAM_{AllPoses}$

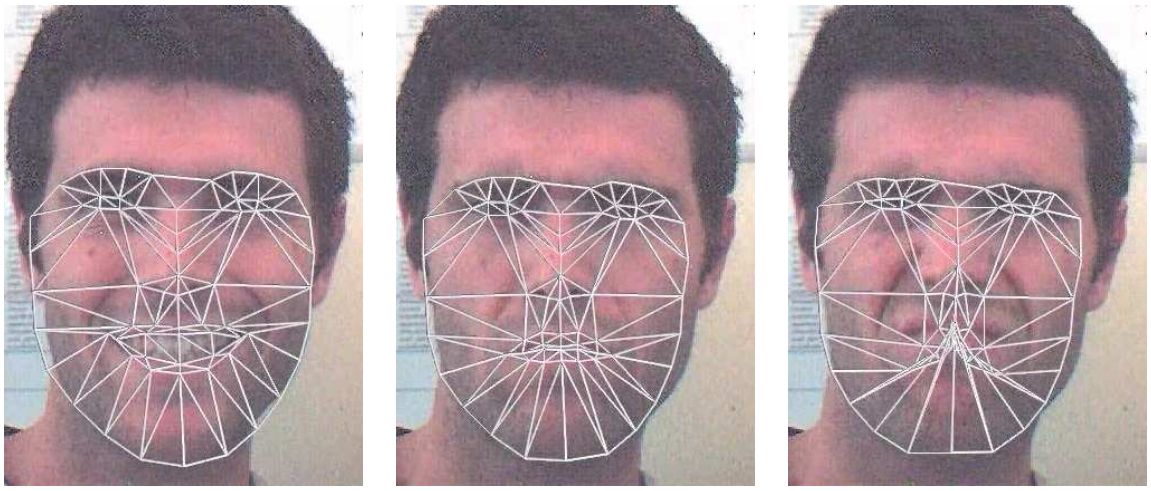
Tracking with $AAM_{AllPoses}$ is launched on the frontal sequence and we first observe its behavior for various amounts of appearance variance (leaving shape variance fixed to 95%), and then its behavior for various amounts of shape variance (leaving appearance variance fixed to 95%).

Effect of appearance variance variation. We mainly observe two different fitting behaviors of the $AAM_{AllPoses}$ when variance of appearance varies. For appearance variance equal or above 80%, the fitting accuracy is visibly good and the stability (robustness) too. Some frames of the tracking can be observed on figure 3.9. For appearance variance below 80%, fitting becomes sensitive to appearance variation and this results in a difficulty to track the face accurately everywhere. The more extreme case of 0% appearance variance (only the mean appearance, with no variation component) is representative of this phenomenon and some frames of the tracking are represented on figure 3.8. For higher than 0% appearance variance and up to 80%, the phenomenon is the same, but weaker as more visual appearance changes can be recovered. The problem desappears for variance above 80%.

To conclude on the effect of appearance variance, its growth allows the model to better represent the face on the image and then to better overlay the model on the face image.

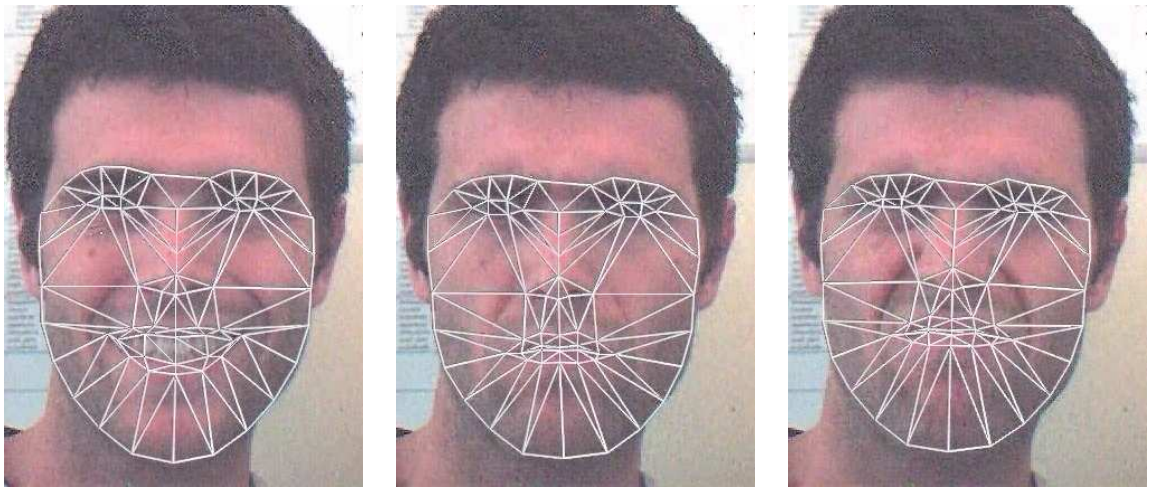
Effect of shape variance variation. The amount of shape variance has an interesting effect on fitting and tracking behavior. We watched the tracking results from the sequence tracked with 0% shape variance to the sequence tracked with 100% shape variance. Several interesting behaviors can be observed.

With no shape variation allowed, the model is obviously rigid: the mean shape is only allowed to transform according to 2D similarities. The fitting is thus very rough and inaccurate since it cannot



frame 29 frame 88 frame 89

Figure 3.8: Tracking result on frontal sequence for the $AAM_{AllPoses}$ with 0% appearance variance (and 95% shape variance). Lack of accuracy fitting is observable on the sides of the face on the first two images: the face appearance cannot be generated completely by the model that therefore compensates the reconstruction error by a shape distorsions then reducing the error. On last image, we observe that this weakness in appearance generability may also be the cause for a lack of tracking robustness.



frame 29 frame 88 frame 89

Figure 3.9: Tracking result on frontal sequence for the $AAM_{AllPoses}$ with 80% appearance variance (and 95% shape variance). The overall tracking quality is very good. Results for more appearance variance appear identical. Fitting robustness between frame 88 and 89 shows that the lack of appearance generability was effectively responsible for the bad robustness when keeping null variance of appearance.

adapt to the evolution of face expressions. In the other hand, the tracking seems very robust. Figure 3.11 illustrates tracking with this model.

When we allow 40% shape variance, the model keeps being quite robust to facial changes, and it does

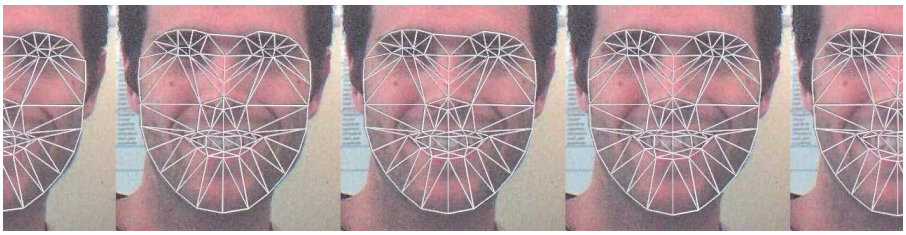
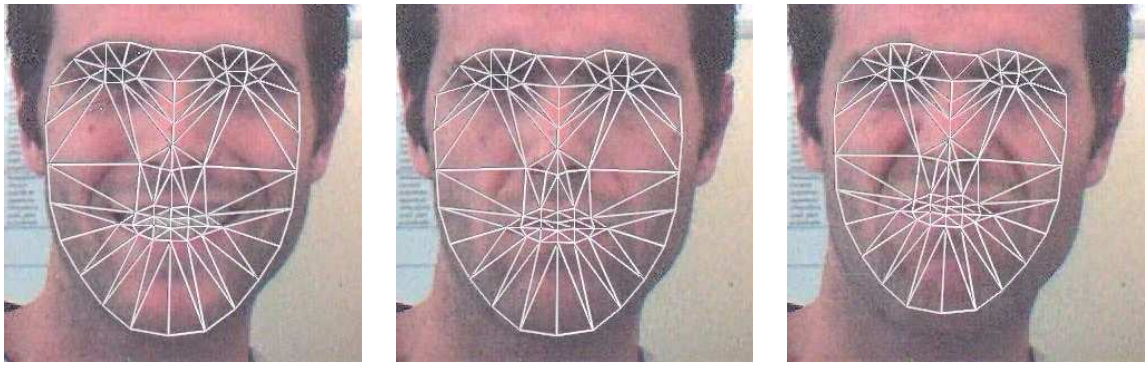
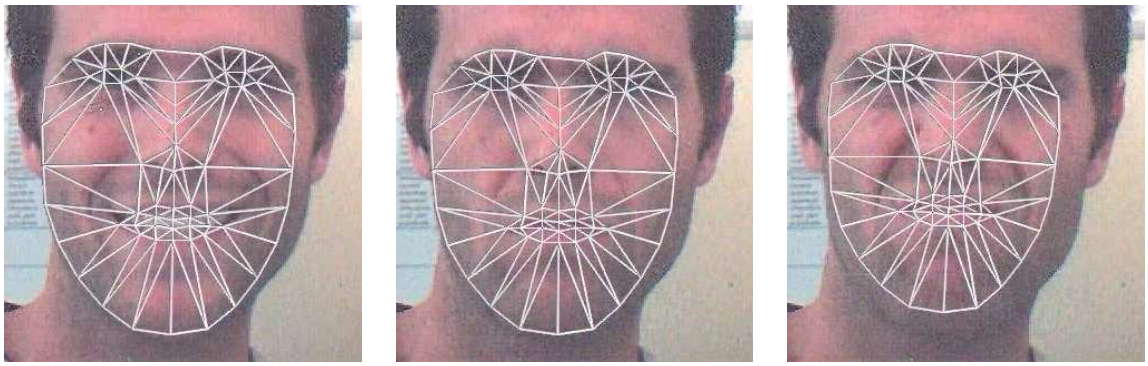


Figure 3.10: Frames 25 to 29 of frontal sequence tracked with the $AAM_{AllPoses}$ parameterized at 95% variance for shape and appearance. The expression evolution is appearantly tracked with good accuracy over the whole sequence.



frame 29 frame 88 frame 89

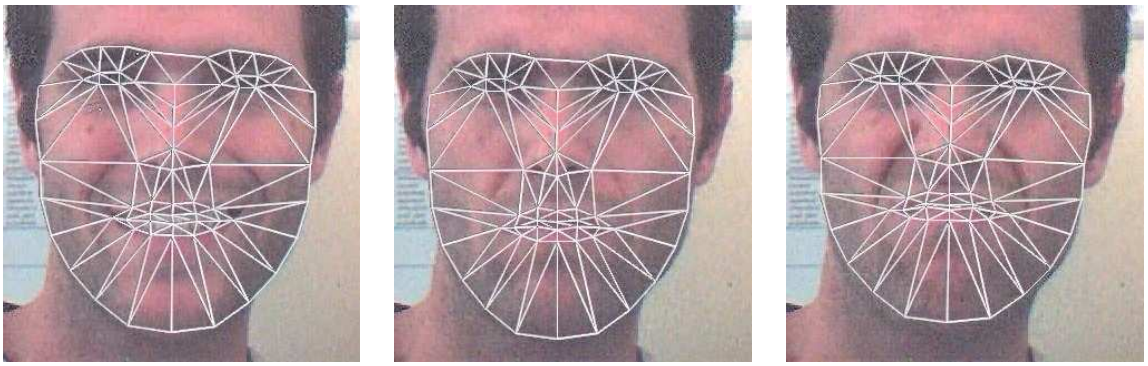
Figure 3.11: Tracking result on frontal sequence for the $AAM_{AllPoses}$ with 0% shape variance (and 95% appearance variance). The fitting is very robust to fast movements but it is also totally rigid (the mean shape is limited to similarity transforms).



frame 29 frame 88 frame 89

Figure 3.12: Tracking result on frontal sequence is shown for the $AAM_{AllPoses}$ with 40% shape variance (and 95% appearance variance). The tracking is robust and rather rigid since only deformations corresponding to head pose movements are allowed.

not deform to expression, but to head rotations instead. The first shape components tend to represent pose changes since these changes generate the highest shape variance in the space representing the



frame 29 frame 88 frame 89

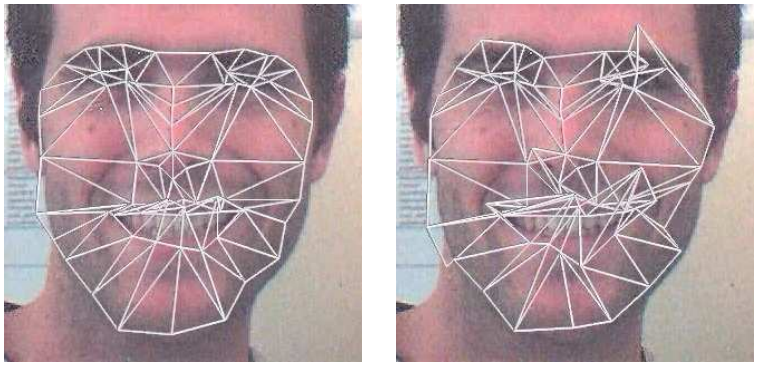
Figure 3.13: Tracking result on frontal sequence is shown for the $AAM_{AllPoses}$ with 90% shape variance (and 95% appearance variance). The tracking is globally good, except on smile where not enough deformation is allowed.

training data. These movements involve all model points to deform largely, and shape data tend to extend themselves along the directions representing these deformations. Other deformations in the training set, such as facial expressions, involve less movement, and are represented with the remaining shape components. Obviously, on the frontal sequence the head is frontal, but the model sometimes assumes this rotational effect to compensate the deformation it is not allowed to operate with so few shape components. The model rotates (very slightly) until error minimization is reached between image and model instance. Figure 3.12 illustrates the tracking with 40% shape variance.

Progressively increasing the amount of shape variance, we allow the model to adapt to effective facial deformations. For 90% shape variance the model can fit almost all facial expressions, except the open smile (see figure 3.13) that is well fitted with 95% and 98% shape variance.

We already know from Figure 3.10 the behavior of smile fitting for 95% variance for shape and appearance. At 98% shape variance the tracking is accurate (maybe more than for 95%) but at the same time it shows a weird shaky attitude and seems to become particularly unrobust to suden variations.

Indeed, tracking robustness is totally lost for 100% shape variance as shows it figure 3.14.



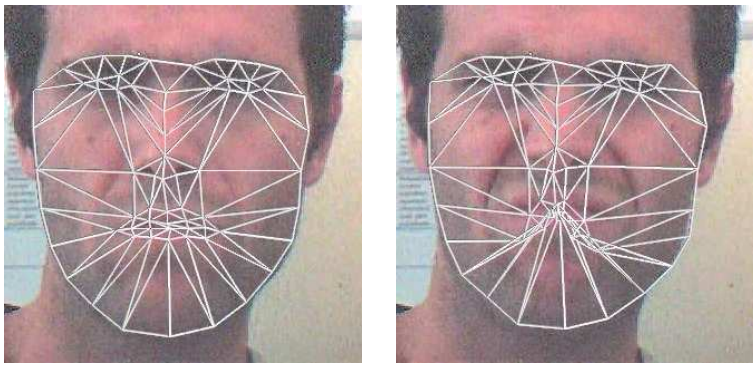
frame 29 frame 38

Figure 3.14: Tracking result on frontal sequence is shown for the $AAM_{AllPoses}$ with 100% shape variance (and 95% appearance variance). The tracker is totally unrobust and eventually diverge.

3.2.2 Track of the frontal sequence with $AAM_{frontal}$

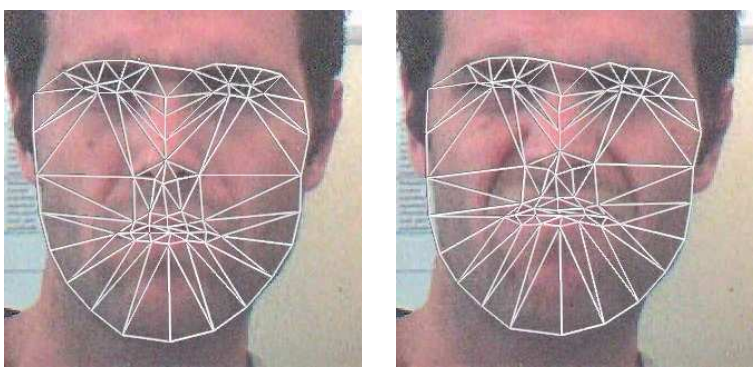
Tracking with $AAM_{frontal}$ is ran on the frontal sequence and we first observe its behavior for various amounts of appearance variance (leaving shape variance fixed to 95%), and then, its behavior for various amounts of shape variance (leaving appearance variance fixed to 95%).

Effect of appearance variance variation. For a null appearance variance, the same troubles as for $AAM_{allPoses}$ are observable: the lack of appearance information creates a mismatch between the model and the image and the tracking can diverge on some highly varying frames. Figure 3.15 illustrates the phenomena. For 40% appearance variance and above, the tracking is performed correctly as illustrates figure 3.16.



frame 88 frame 89

Figure 3.15: Tracking result on frontal sequence is shown for the $AAM_{frontal}$ with 0% appearance variance (and 95% shape variance). The tracker presents a lack of robustness due to its unability to retrieve properly all visual particularities of the face on the image.



frame 88 frame 89

Figure 3.16: Tracking result on frontal sequence is shown for the $AAM_{frontal}$ with 40% appearance variance (and 95% shape variance). For 40% appearance variance and above the tracker performs correctly.

Effect of shape variance variation. For 0% shape variance, the tracking is robust and rigid, as observed for the $AAM_{allPoses}$. For 40% shape variance, all deformations can be retrieved by the model,

except for the smile. For more shape variance, the tracking is done accurately. 100% shape variance does not make the tracker loss its robustness.

3.2.3 Further comments

As a first comment, we must say that an AAM (when properly parameterized) has the property to generate the images of a whole sequence when it has learnt only some key-frames of the sequence. Not learnt images are similar to those used for training in the sense that they represent the same expressions at lower intensity. We observe on sequence tracking that the fitting is accurate for the whole expression evolution.

Good tracking quality is obtained with 95% variance for shape and appearance, or 98% can suit too. These are the quantities that are commonly applied by AAM users to train their models. We will see that in the case of fitting on unseen data, the variance quantities that optimize the fitting performances can be different.

3.3 The person-specific context: test of performances

The observation of tracking on the frontal sequence has led us to distinguish three main performance criterion that we would like to explore more thoroughly. A change on shape variance potentially leads to more or less fitting accuracy. **Accuracy** is the first criteria. For a given variance parameterization of shape and appearance and depending on the face displacement from a frame to other, we could observe a certain level of tracking robustness. The second criteria is the fitting **robustness**. Unlike the terminology belonging to statistics, robustness here should not be viewed as resistance to noise, but we consider it to be the resistance to shape perturbation, or to initial bad initialization. In this way, characterizing the robustness of an AAM in the sense we understand it is equivalent to measure the width of the convergence basin.

The last criteria is the **speed** for convergence. According to the amount of learnt data and the amount of components retained, the fitting process can be slower or faster in terms of two things: the processing time required per iteration, and the number of iterations necessary to converge.

To explore these three criteria, we will operate an exhaustive fitting test on a single image taken from the training set.

3.3.1 Fitting accuracy

We retain the frontal open smile face image for this test. This image, extracted from the training set relative to the frontal sequence, has been manually labelled and we will use these labels as ground-truth to score the fitting. We will call *reference shape* these ground-truth labels, and *reference appearance* the appearance that lies inside the reference shape.

We want to test the potential accuracy that can be reached by an AAM according to its shape and appearance variance parameterization, and to the amount of data it learnt. We will run the fitting of $AAM_{frontal}$ and $AAM_{allPoses}$ onto the test picture. There are different manners to initialize the fitting process. If we start the model far from the solution, there is a risk that it gets stuck into a wrong minimum. The ability to avoid wrong local minima more concerns the robustness criteria that will be test consecutively. To evaluate the potential accuracy that can be reached by an AAM, it is convenient to give the model the closest initialization to the best solution it can reach. In this way we increase the chance to lead the fitting process onto the global minima. The distance to the reference shape when the model is in the global minima corresponds to the fitting accuracy. We want to see its

dependency to the shape and appearance parameterization as well as to the amount (and nature) of training data.

A very good initialiation of the model on the picture is given by applying to the model the shape and appearance parameters retrieved as follows. The shape parameters are obtained by projection of the reference shape onto the shape subspace of the AAM. The appearance parameters are also set to the values obtained by projection of the reference appearance onto the model appearance subspace.

It is possible to compute the error fitting at each iteration to have an idea of how far the model vertices are from the optimal position. As a distance measure we use the simple point-to-point error, or distance, defined as the mean of the euclidean distances of all model vertices to the ground-truth position given by the reference shape.

When the point-to-point error does not vary anymore, the fitting has converged and the final error represents the fitting accuracy for a given model ($AAM_{frontal}$ or $AAM_{allPoses}$) and a given variance parameterization.

On figure 3.17 a curve presents the accuracy that can be reached with the $AAM_{allPoses}$ for various percentage amounts of shape and appearance variance retained. To give a visual idea of how the fitting looks like for the represented point-to-point errors, figure 3.18 associates visual fitting results for some points of the curve.

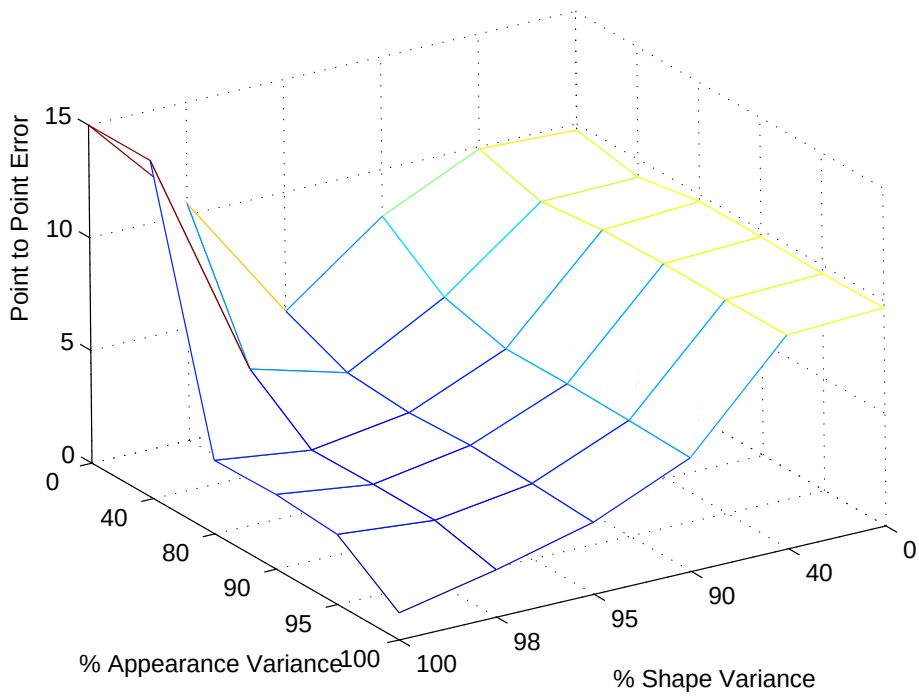


Figure 3.17: Best fitting accuracy obtainable with $AAM_{allPoses}$ for each couple of shape and appearance percentage variance tested.

This curve contains a dense information that will be completed by the test of robustness.

For a given amount of shape variance, the best accuracy is obtained when more appearance variance is retained. In practice, from a certain amount of appearance variance, the fitting accuracy does not improve much: for 80% and above, the accuracy remains globally the same (a slight improvement is

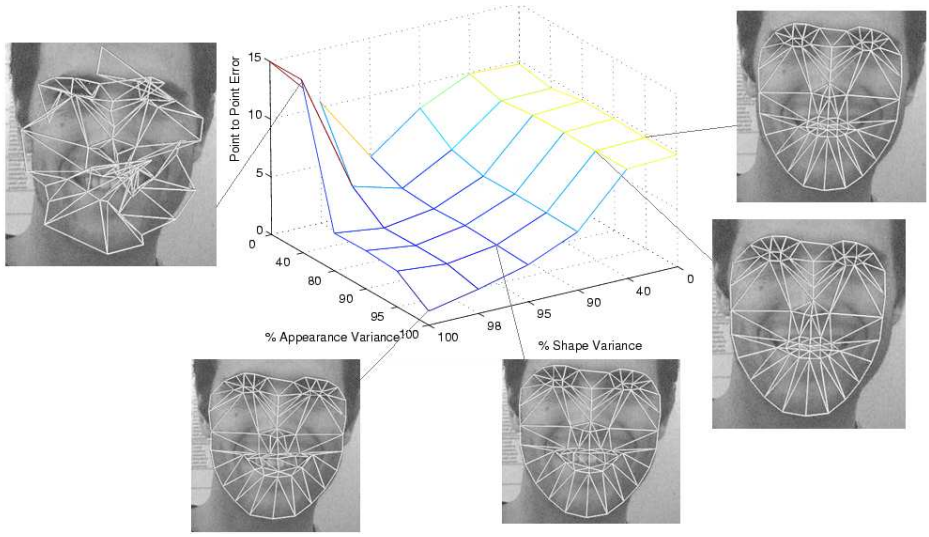


Figure 3.18: The same curve is shown illustrated by visual fitting results that correspond to some main fitting scores.

observed at full shape variance).

For less than 80% appearance variance, a curious phenomena happens when enough shape variance is retained: the fitting diverges completely, even when starting from the optimal position. A highly deformable model that cannot express much visual information of the face image results in a lack of robustness as shown in the singular area where fitting diverges on figures 3.17 and 3.18.

The explanation is the following: the model cannot explain the image entirely (low appearance variance), then an error will always remain between image and model. If the model cannot deform much (low shape variance, see on the curve), the fitting will be coarse but will not diverge since it is constrained by its non-deformability. When it is more deformable (high shape variance), the model is sollicitated to deform in order to minimize the remaining error between the model and the image, leading to unplausible shapes.

For any amount of appearance variance above 80%, $AAM_{allPoses}$ potentially improves its fitting accuracy when increases the amount of shape variance. The more deformable the model, the closer it can get to the reference shape.

To conclude for $AAM_{allPoses}$, when enough appearance variance is available, the more deformable the model is, and the more accurate it can potentially be. We will see that when the model is highly deformable, it is also less robust to initial perturbations.

As shows figure 3.19, $AAM_{frontal}$ is subject to the same rules, except that the divergence area is not present.

3.3.2 Fitting robustness (convergence basin)

The fitting robustness is measured for several intensities of the model starting position perturbation. From a given starting position, the model - if it is robust enough - should converge to the position it assumes when it is launched from the optimal parameterization as it was the case in the previous test

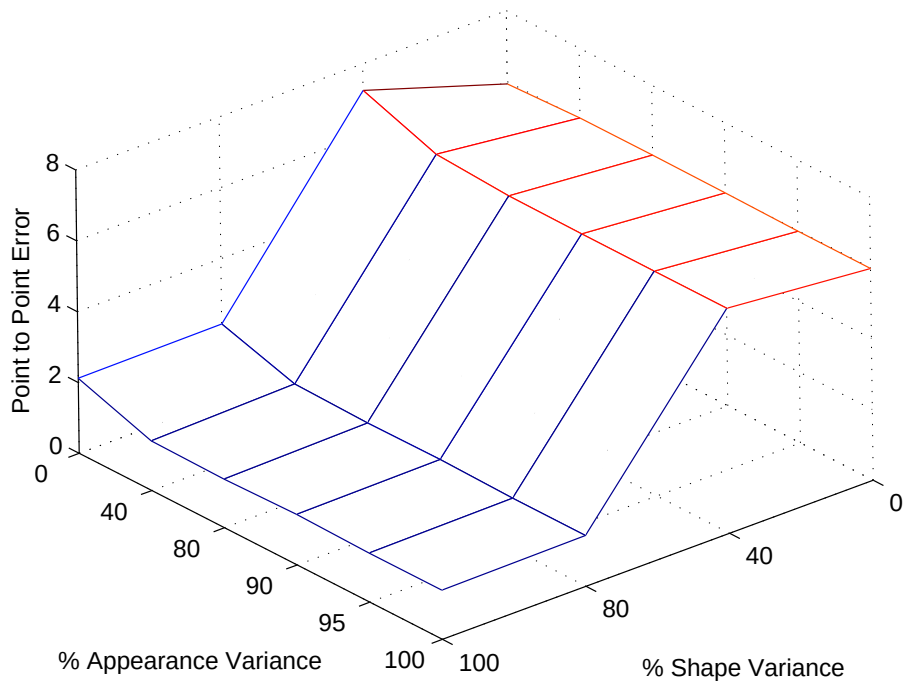


Figure 3.19: Best fitting accuracy obtainable with the $AAM_{frontal}$ for each couple of shape and appearance percentage variance tested.

on fitting accuracy. These optimal positions are indicated by the point-to-point distances reached for a given parameterization of the model shape and appearance variance (see figure 3.17).

To generate perturbated initialization positions on the model, we first project the reference shape onto the shape subspace composed of the similarity transform components only (no deformation component is considered). We thus obtain the shape parameters to apply on the (mean shape) model to position it on the image at the closest possible position from the reference shape. In a second time, we randomly perturb the similarity transform parameters of this model mean shape to bring it onto a new starting position on the image.

The point-to-point distance between this mean shape and the reference shape is computed for the randomly generated perturbation. This distance is rounded to the closest integer value, and the starting position given by the mean shape position is stored in a bank and indiced with the rounded point-to-point distance value that will be called *starting distance*. We then obtain a bank of starting positions of increasing difficulty.

For each variance parameterization (we take the appearance variance higher than 80%) both AAMs are ran from the same 20 starting positions for each starting distance we wish to test. The fitting process is launched for a number of iterations that allows convergence whenever the fitting can converge. With respect to the optimal position reached by the model for optimal initialization, we consider that the model has converged when it stabilizes on the same point-to-point error, with a tolerance set to ± 0.5 pixel. On 20 trials, we compute the percentage of effective convergence.

For each starting distance we test, we represented the percentage of convergence for all possible variance parameterizations (with appearance variance set above 80% to avoid the useless divergence

area).

For the starting distance equals to 10, 12, 15, 18, 22, 26 we test the percentage of convergence of the $AAM_{allPoses}$. Some starting positions are illustrated on Figure 3.20. For each starting distance we represent the percentages of convergence for all combinations of shape and appearance variance tested. Results are reported on figure 3.21. The main observation is that the higher the shape variance, the more sensitive is the robustness when initial error increases. For 100% shape variance, the robustness is extremely poor.

For low shape variance, the model is robust to initial perturbations, and this is particularly true for the rigid (null shape variance) model that converges also for extreme initial perturbations. For high shape variance, the model can deform more and lose its robustness to initial perturbations.

The explanation of this phenomenom stands in the following. The model perturbation of the model from the solution gives an error image which usually presents more error when the perturbation is high. This error is used to compute the shape update to apply on the deformation components.

When few shape components are retained, the shape variance they represent corresponds to high and global facial shape deformations present in the training set. Thus these components are well adapted to conduct the model into a better position on the input picture. A large error in the error image will lead to faster replacement on the picture.

When a large amount of shape components is retained to express all kind of shape variance, the large variance due to high and global deformations (represented by the first shape components), and the small variance due to subtil deformations (represented by the last shape components), a robustness problem can be observed when the initial perturbation is high. A large error on the error image unadaptedly activate the small variance shape components. Those components unappropriatedly answer to this error stimulation with a large displacement of the model vertices, leading to an uncoherent and diverging fitting. It has actually no sense to use the small variance shape components when the model can be far from the solution. The small variance components should only be used to refine the model position when it is already very well located on the picture.

The effect of appearance variance is not easy to interpret and we will consider that appearance variance fluctuation for values equal or above 80% has no particular effect on fitting robustness.

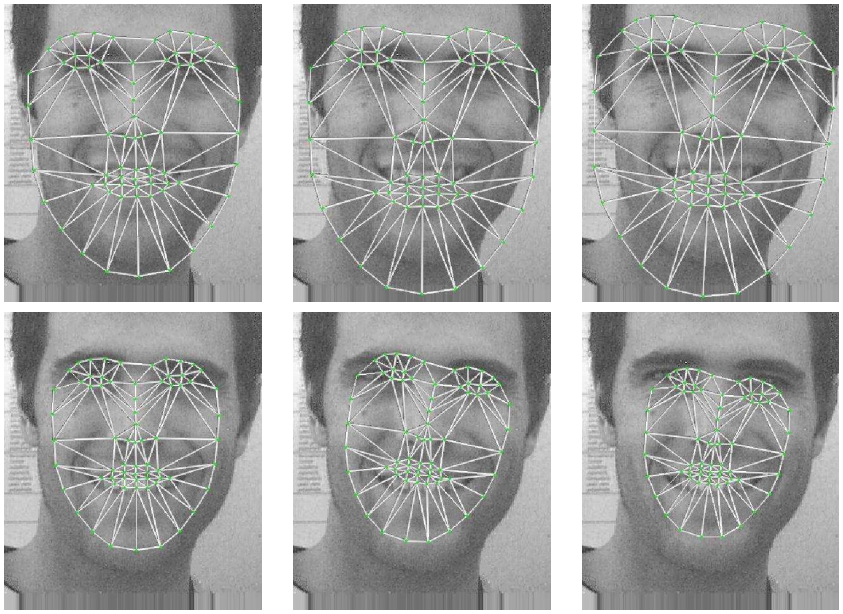
3.3.3 Fitting speed

For a given amount of training data, the number of shape and appearance components correponding to the quantity of variance of shape and appearance have an influence on fitting accuracy, robustness to initial perturbation, and also fitting speed. Speed, the third evaluation criteria of the AAM fitting performance will be now studied.

For all shape and appearance percentage of variance tested, we perform a same tests as for robustness, but this time, scoring the fitting accuracy at each iteration. For each starting distance tested, the same 20 trials are performed.

It is interesting to visualize through iterations the moment in which an AAM fitting converges to the optimal position (always given by the final position reached by the AAM when ran from the optimal parameterization). We retain the fitting trials that converged to the optimal position and propose to visualize their average rate of convergence. We display those rates of convergence as follows: for a fixed amount of appearance variance, we display the rate of convergence for all shape variances and starting error positions tested. Figure 3.23 shows the rates of convergence for each appearance variance above 80% tested for the $AAM_{allPoses}$. Figure 3.24 shows the rates of convergence for each appearance variance tested for the $AAM_{frontal}$.

For both $AAM_{allPoses}$ and $AAM_{frontal}$, increasing the appearance variance also slightly increases the number of necessary iterations to reach the convergence. The computational cost also increases



15pxl perturbation 22pxl perturbation 26pxl perturbation

Figure 3.20: Illustration of some starting distances used to test the frequence of convergence (the robustness) of both $AAM_{allPoses}$ and $AAM_{frontal}$

for each iteration because of the higher number of components. For appearance variance above 80% we saw that the benefit of accuracy fitting and fitting robustness is modest.

It is then important to retain above 80% variance to avoid the lack of robustness observed when low amount of appearance variance and high amount of appearance variance are combined to build the AAM. However, above 80% variance of appearance any amount is fairly suitable for person-specific tracking applications, and it may be interesting not to rise it too much to save some computational time.

For both $AAM_{allPoses}$ and $AAM_{frontal}$, increasing the shape variance potentially increases fitting accuracy and lessens fitting robustness (not in a noticeable way when few data are learnt), and the fitting process is slowed down in terms of number of iterations neccessary to reach the convergence, and in terms of increased computational cost per iteration.

3.3.4 Comments on the person-specific context

We now compare AAM accuracy, robustness and speed when the amount of data increases: what differences are observed between $AAM_{allPoses}$ and $AAM_{frontal}$?

For any quantity of learnt data an AAM presents the same kind of response when shape and appearance amount vary: accuracy potentially increases when shape amount increases, while robustness potentially decreases and speed too because of the higher complexity of the AAM. As about appearance variance, it must be maintained above 80%, value above which the behavior do not vary much.

However the less data are learnt, the better the overall performances of the model. Both AAMs can potentially present the same fitting accuracy. However the $AAM_{allPoses}$ needs more shape variance than $AAM_{frontal}$ to reach an equivalent accuracy, what means an increase of computation time per iteration and number of iterations required to reach the solution, and a lower robustness to perturbations from the correct solution.

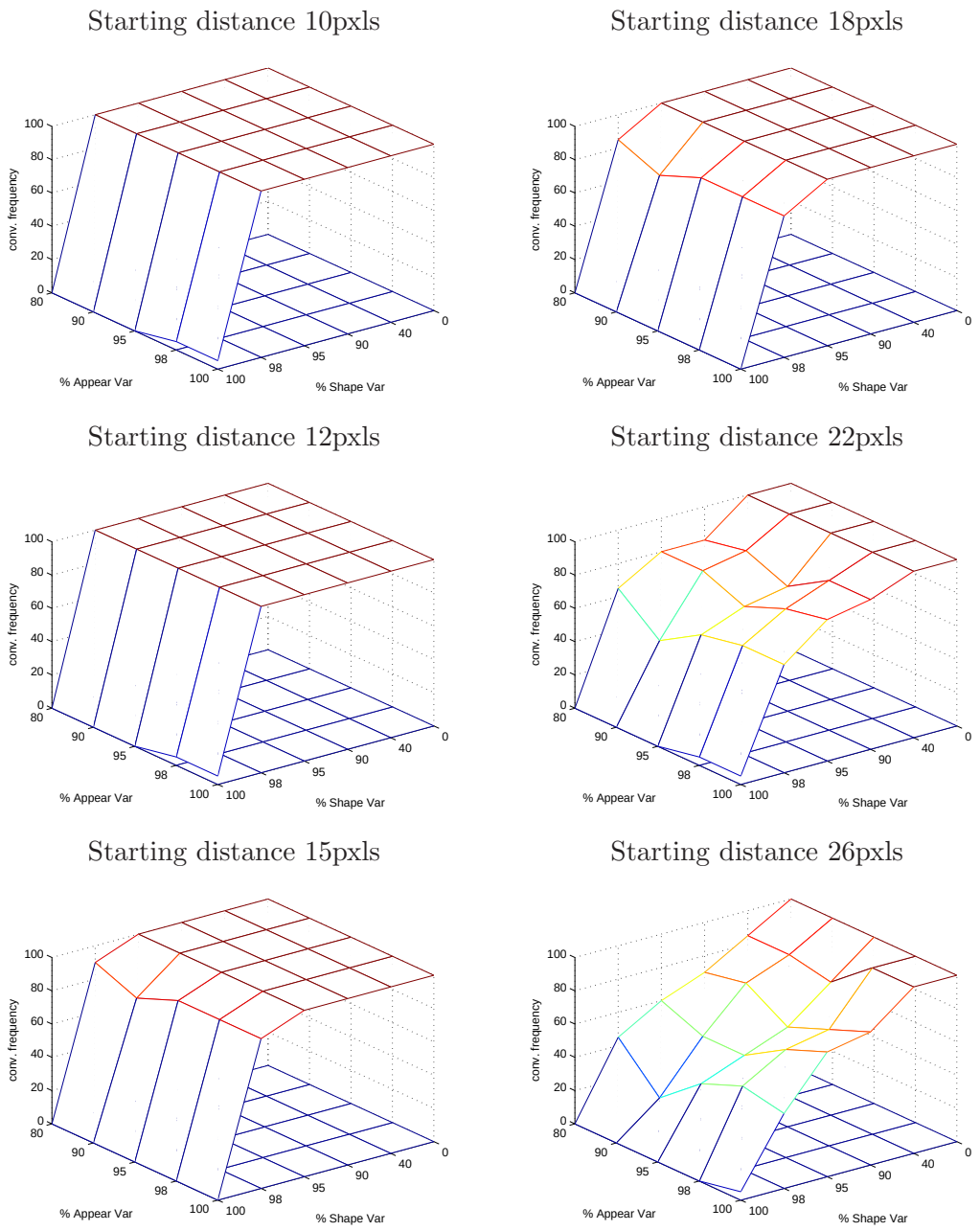


Figure 3.21: Frequencie of convergence of $AAM_{allPoses}$ for different starting distances. For each starting distance, 20 trials are launched, and they are repeated for different settings of shape and appearance percentage tested. The rise of shape variance retained to build the model shows to lessen its robustness to initial perturbations. For small shape variance, the model keeps robust. The large model displacement makes a large error image. This error used to compute the shape update to apply on the first deformation components (representing large training data variance) often leads the model to converge. However the same error applied to the last components (representing small training data variance) when these are retained in the model leads the model to deform unappropriatedly and causes the model to diverge.

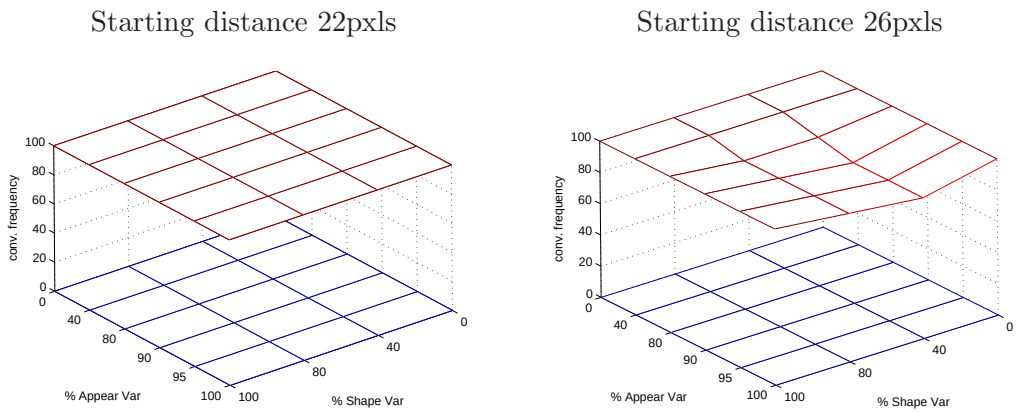


Figure 3.22: Frequence of convergence of the $AAM_{frontal}$ for 20 trials at different starting distances, and for different settings of shape and appearance percentage to build the model. This lighter model keeps a very good robustness even for large initial perturbations.

3.4 Conclusion

The main conclusion of the chapter basically stands in that the observations of [75] comparing a single-person AAM with a 5-people AAM also holds when we increase the amount of data of the single-person AAM: the robustness and speed of the AAM decrease when more data are learnt. We went through a complete test of the AAM setting parameters in the person-specific context. The goal was to provide a very complete understanding on the way the process works in this context. It gives a basical knowledge to get into the remaining chapters.

We do not aim at performing an extensive test of the single-person case here. However, if one plans to train a person-specific AAM on an extensive amount of poses and facial deformation of one person, then the AAM robustness might decrease to a point which is not acceptable for real applications. Several solutions to such a problem can be proposed:

1. the constraints placed on shape and appearance variation as proposed in the extension [5]. This can present the advantage to prevent from many local minimas, but it increases even more the computational cost per iteration and does not prevent against the long search process (high number of iterations required) to reach the solution.
2. the shape variance can start low, and can progressively be added to the AAM to start with a robust but inaccurate fitting and progressively tends to more fitting accuracy while decreases the robustness that becomes unecessary when we get close to the solution. We conducted an experimentation on this solution, but the exit was not very encouraging. It is hard to define an automatical criterion to increase the amount of shape data at the right time during the fitting process. Waiting too long might block the AAM onto a minimum from which it will not go out, whereas changing too soon may give it too much deformability too soon and drive it into a wrong solution.
3. the data to be learnt by the AAM can actually be partitioned over several AAMs, for example learning all facial deformations for only one pose. This fewer complexity in each AAM of the built collection would then perform faster and with more robustness than a heavy AAM enclosing all training data. The problem would consist in selecting from the collection, the AAM that best suits to the current frame. This can be done strategically by testing on the current frame the same AAM and all adjacent AAMs to the one used to fit in the previous frame. The winner

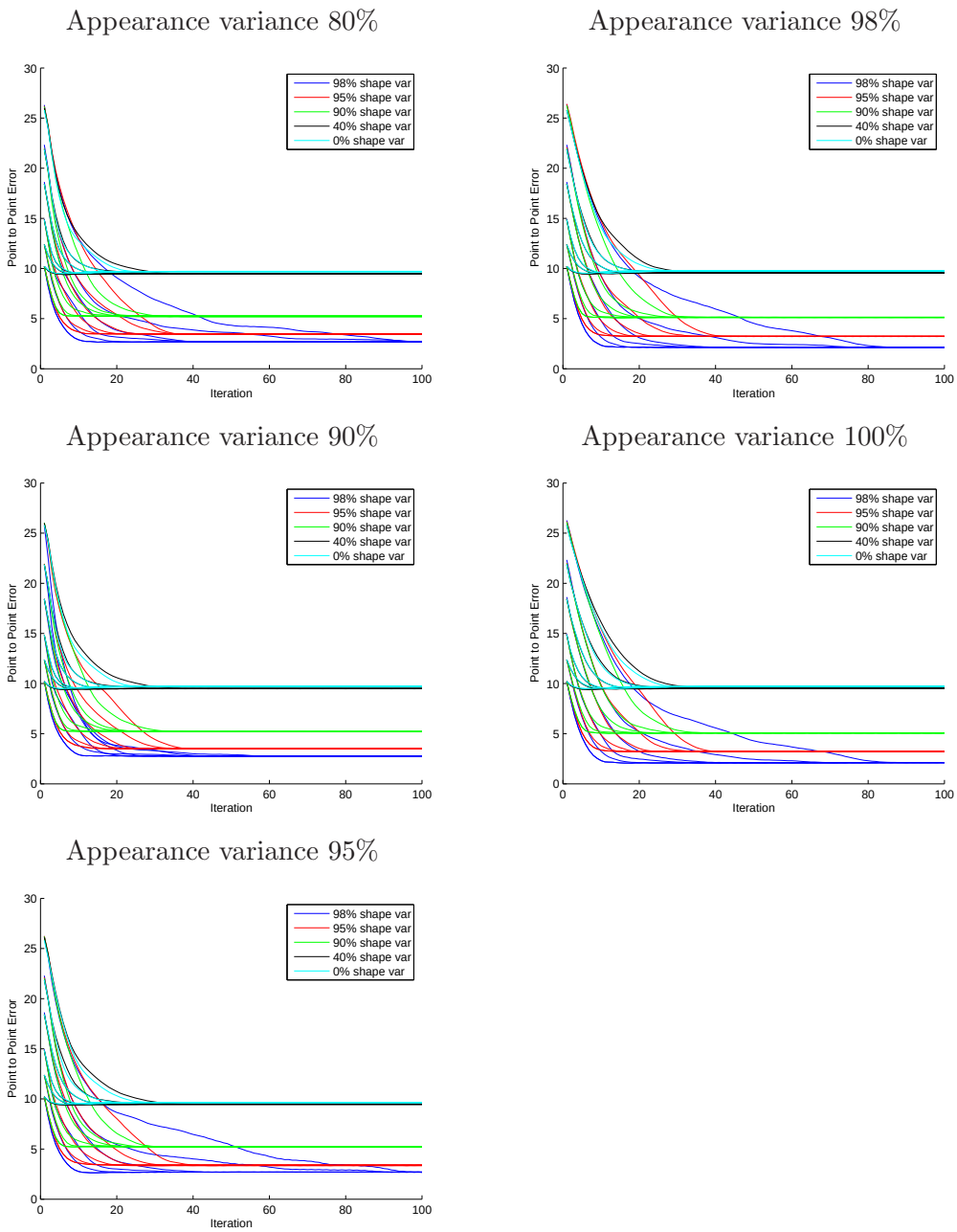


Figure 3.23: Rates of convergence for the $AAM_{allPoses}$. More shape or appearance variance retained lead to a longer fitting process in terms of iterations,and the iteration is also longer to compute. The potential accuracy is higher when more shape and appearance variance are retained.

among all AAMs tested can be the one that presents the lowest average error fitting.

We believe that solution number 3 can be reliable and this will be confirmed by the experiments done in the next two chapters. Moreover, it offers a very useful property. Pose and expressions information are intrinsically combined when everything is stuffed inside one AAM. In [110], structure from motion is used to retrieve the corresponding 3D shape of the AAM, then allowing to separate the pose from the expression. Training one person-specific AAM per pose, then only learning the

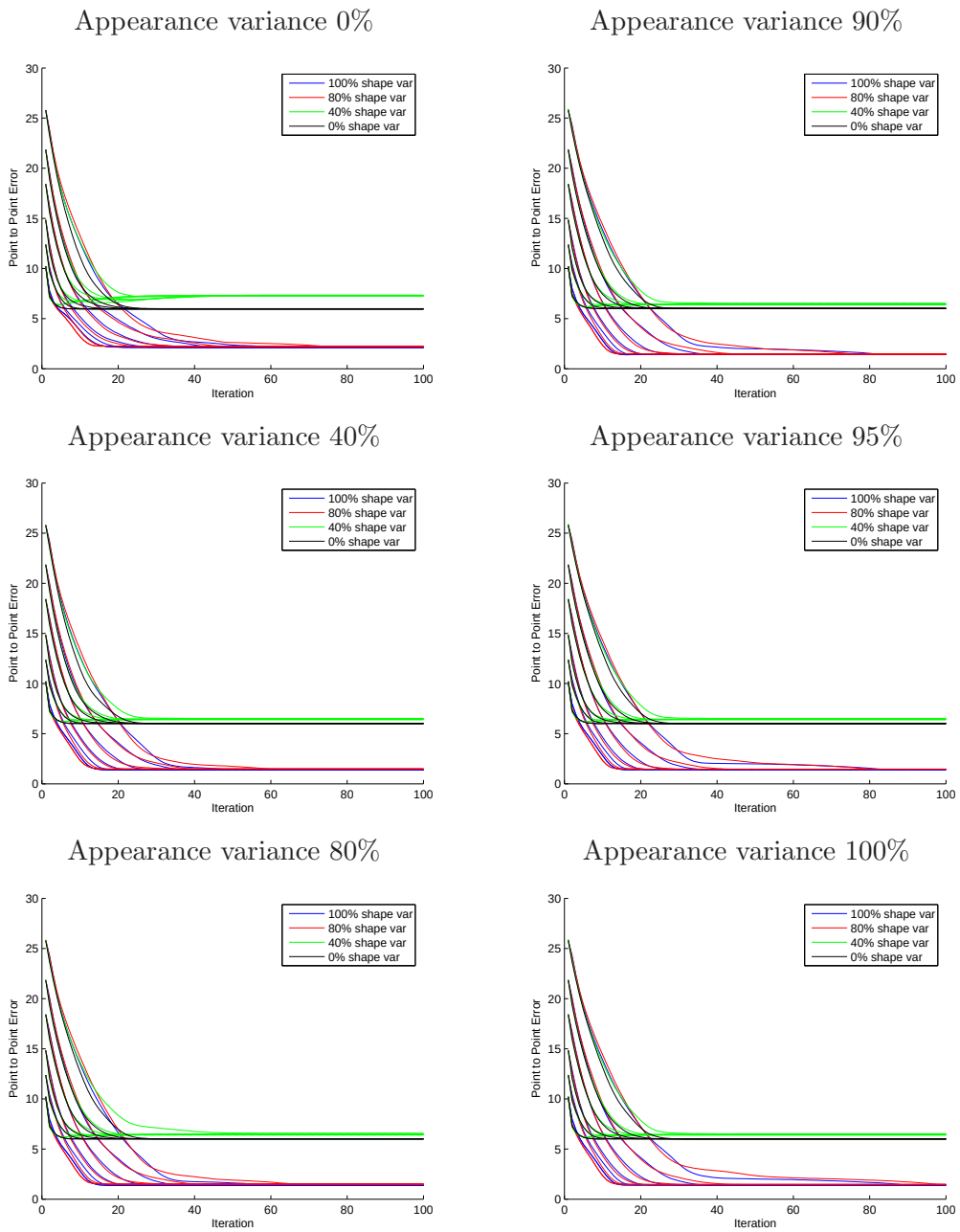


Figure 3.24: Rates of convergence for the $AAM_{frontal}$. The same as for the $AAM_{AllPoses}$ comments can be done, however saying that except for the case of 0% of appearance variance that presents a smaller fitting accuracy, the difference between each set of curve is pretty small.

expression, would actually desambiguate the pose separation in a more simple way, and then only the expression should be retrieved. This can be done in a simple way through the comparison of the current fitting to the different expressions for the given pose. This obviously implies that all training data have been previously labelled in terms of pose and expression it represents.

This latter solution makes the person-specific context available and reliable for building applications. It is still to know whether they could run in real-time, but several implementations have been proposed

for the person-specific and for more generic contexts, that work in real time [64, 112].

We will see in next two chapters that this strategy consisting to build a partition of the data to be learnt into several AAMs to specialize them on particular tasks, therefore optimizing their performance, can also be a solution to better face the difficult problem of fitting on unseen faces.

Chapter 4

Fitting Unseen Faces

4.1 Introduction

Fitting an AAM onto an unknown/unseen face with accuracy seems to be a considerable challenge. In the person-specific context the fitting robustness decreases when increases the amount of learnt data. The generalization to new people would actually require the AAM to be trained onto a large variety of identities, under a large range of poses and expressions. We can expect the fitting robustness to be significantly low in that case, making the AAM useless with the classical gradient-descent fitting solution, too prone to get stuck into local minimas. This observation lessens our hopes that the method can easily generalize on unseen people.

However the system we would like to build should be able to deal with unseen people. Furthemore the question of accurately fitting an unseen face represents the key-vault for many applications and is still an open problem today. It is then of major importance to understand more about the way AAMs perform in this context.

We observed that an AAM trained on a small amount of data can perform robustly and efficiently for the images it is trained for. I decide to investigate the unseen context, this latter observation leads us to conduct a first investigation under a context where the amount of learnt data is restricted. We then maximize the chance to observe something working in this context. In this chapter, we propose to investigate the AAM *generalisability* on unseen faces (the identity is unknown to the AAM) under one fixed pose and one fixed expression (illumination is also fixed to homogeneous). The pose is fixed to frontal and the expression to neutral. This reduces to the minimum the amount of data that the AAM must effectively learn in the case of unseen data. We thus can expect the results in this restricted unseen context to be the best results we can obtain in the unseen context.

Although the context of investigation is restricted, such a capability is desirable for many automatic applications. Moreover, if this investigation provides a positive exit (as we will see it does), we will not expect positive results for a growing range of pose and expressions stuffed into a unique AAM (since it would lose its good performances), but instead we can expect to build up a solution based on multiple models, each one specialized for one pose and one expression. The question would then be to find how to select the model that best suits to the current pose and expression, and switches from a model to other during the face tracking. We will start to investigate this strategy in chapter 5.

We will study the AAM fitting behavior when it is trained on frontal and neutral faces and ran on the same kind of images where faces are unknown to the training set. Limitations of fitting on unseen faces will be highlighted and explained all along §4.2. A new, more objective fitting accuracy measure will be introduce in §4.2.1. The explanation will become clear while observing the fitting behavior when the face is known to the training set: the AAM appearance counterpart limits the generalisability and the fitting accuracy on unseen faces. To overcome the problem, we propose a

solution based on local/segmented models. They cover a smaller area of the face, then reducing the appearance space dimension, what increases the appearance statistics expressivity. Local models shape and appearance generability will be tested and compared to global models' in §4.3. Local models demonstrate better generability performances, and their use lead to more accurate fittings. We will first start the discussion with a reconsideration of a very common problem, which is usually unconsidered in literature: how to objectively judge upon the accuracy of a (manual or automatic) fitting on an unseen face?

4.2 Why is fitting accuracy limited on unseen faces?

4.2.1 Statistical Shape Error

When we fit an AAM on a known/seen face as we did in previous chapters, it is legimate to judge the fitting accuracy with respect to the manual labelling that were used for training: the model learns this configuration and corresponding appearance, it should then be able to reconstruct it. In practice this is not completely true since the PCA removes signal high frequencies when we retain a bit less than 100% of the training set appearance variance: it happily happens that some undesirable noise due to manual errors is eliminated this way, and an automatic fitting can usually result of better quality than the original manual labelling.

Whereas manual labels can reasonnably be acceptable for judging fitting accuracy on learnt data, it becomes problematic in case of fitting on unknown data. To test the quality of an AAM fitting onto a face image, methods other than visual control are difficult to build. A common mistake consists in benchmarking the fitting result against a manual labelling: it is incorrect to claim that a manual labelling is objectively better than another one.

As a solution to this judgement problem, we introduce a statistical-based method to build the ground truth data. A fitting error is given to a labelling, either manual or automatic, taking into account the degree of accuracy of human experts to manually localize each AAM vertex.

To define the ground truth shape and the fitting error, we rely on a high number n_L of expert labels for each of the n_I face images. In this way, a probability distribution can be computed on each of the n_V vertices. For each image i, the mean $\boldsymbol{\mu}_{i,v}$ of each vertex v is computed over its n_L labels, defining the ground truth shape $(\boldsymbol{\mu}_{i,1}, \ldots, \boldsymbol{\mu}_{i,n_V})$. A covariance matrix $\boldsymbol{\Sigma}_{i,v}$ is computed over the n_L labels for each vertex v of each image i, considering the distance of each label $\boldsymbol{x}_{i,v,l}$ to its corresponding mean $\boldsymbol{\mu}_{i,v}$.

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

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

A spread distribution (*i.e.*, having high variances σ_x and/or σ_y) means a vertex hard to localize or not well-defined. A concentrated distribution (*i.e.*, having low variances) means a vertex well-defined. The idea is to penalize less an inaccurate vertex localization (either manual or automatic) when manual labelers were inaccurate to define its position (spread distributions of the n_L labels), and to penalize it more when manual labelers were accurate. Figure 4.1 shows face images overlaid with their manual labelling statistics, with each vertex represented by an ellipse showing its mean position and uncertainty.

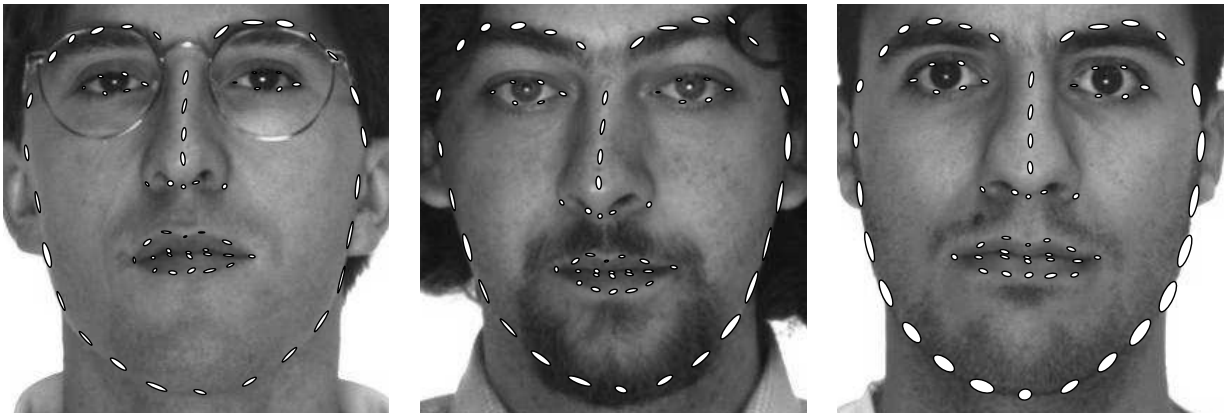


Figure 4.1: Three of the 40 faces that were annotated 10 times. Covariance ellipses represent the distribution of the 10 labels around mean vertices.

The fitting error $SSE_i(\boldsymbol{s})$ of a shape $\boldsymbol{s}$ on an image i, is defined by the average of the Mahalanobis distances:

$$SSE_i(\boldsymbol{s}) = \frac{1}{n_V} \sum_{v=1}^{n_V} \sqrt{(\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})^T \boldsymbol{\Sigma}_{i,v}^{-1} (\boldsymbol{s}_v - \boldsymbol{\mu}_{i,v})} \tag{4.1}$$

where $\boldsymbol{s}_v$ is the vth vertex of the shape $\boldsymbol{s}$.

We collected 40 images from the AR database ([73]) containing only frontal faces, without expression and with global fixed illumination. To build our training set, we manually labeled 10 times (to define the ground truth shape) each image with a landmark configuration always describing the 68 vertex mesh used in [75].

4.2.2 Fitting unknown faces

In [56], Gross, Matthews and Baker started to address the problem of fitting onto a new frontal and neutral face, announcing that a more complete version of this paper was to appear. In the meantime while we needed certain answers to carry on the research and we were precisely investigating on the problem, these authors published [57], the announced version of the paper. Whereas no one investigated on this aspect before, in [57] authors highlighted the effect of the amount of shape and appearance variances on the fitting quality on unseen faces. In their tests, the convergence is claimed on unseen faces when the fitting error is less than 2 pixel distance from the manual labelling given as ground truth. The authors observe the frequency of convergence and determine a setting for the shape and appearance variance to retain that maximize the number of convergences. However they do not explain why this number is limited.

Our tests are somehow more accurate since we do not decide binarily if a fitting has converged or not, but we measure the fitting accuracy for all available images as objectively as can allow our SSE measurement method.

This test aims at observing what happens when fitting onto an unseen face, with a varying amount of shape and appearance variance retained.

The test is done in a leave-one-out manner: among the 40 available images, 39 are extracted from the pool and are used to train the AAM. The model is fitted onto the remaining picture. We then do 40 tests. For each tested face picture, we ran 196 fittings, each time recomputing the number of components of shape and appearance retained, in order to span the whole range of possible variances

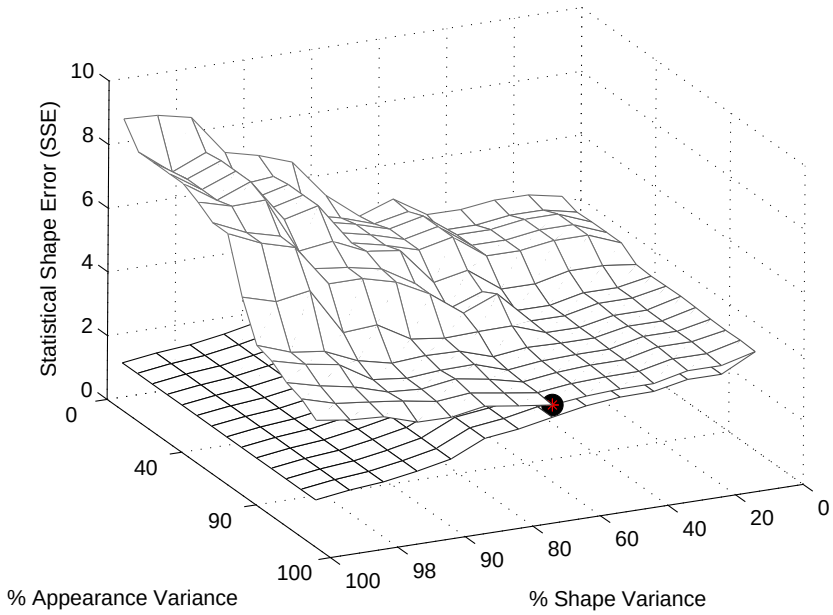


Figure 4.2: *Unseen* contexts analysis. The bottom curve represents the initial SSE of equation (4.1) averaged over all the 40 images. The top curve shows the average SSE after the fitting process has ran. Various amounts of shape and appearance variances are tested. The black dot represents the point of best average fitting accuracy on unseen faces observed for 60% of the shape variance and 100% of the appearance variance.

retained to build the AAM. The fitting process is initialized from the ideal possible parameterization obtained by projection of the test face onto the shape space to retrieve the initial shape parameters and onto the appearance space to retrieve the appearance parameters. In this way, we ensure that if the model diverges, it is not due to potential local minimas but to the true model inability to fit a given unseen face.

For each fitting trial, we score with the SSE the fitting accuracy of the AAM before the first and after the last iteration of the fitting process. Since the AAM is initialized in the best possible position, the SSE cannot be better than it is at initialisation. What we want to observe here is whether the fitting process leaves the AAM into its best, initial position, or if something leads it to leave this position, then consequently losing some fitting accuracy. We run the fitting process over 60 iterations.

The average SSE scores obtained over the 40 trials for each couple of shape and appearance tested are reported on figure 4.2. These SSE results are reported on figure 4.2. The bottom curve represents the (lowest possible) SSE scored at fitting initialization in average over all 40 trials, and for each couple of shape and appearance variance tested. The top curve represents the SSE scored at the end of the iterative fitting process, also in average over all trials and for each tested combination of shape and appearance variance.

In [57], authors show that the overall best performance in terms of *fitting frequency* is obtained for 90% variance in shape and 97% variance in appearance. This setting of the AAM does not eventually lead to a major SSE difference with the lowest overall SSE we observe for 60% of the shape variance and 100% of the appearance variance retained. However this difference can be explained by the following. The fitting error of the non-converged trials is *not* taken into account in [57]'s test. Since

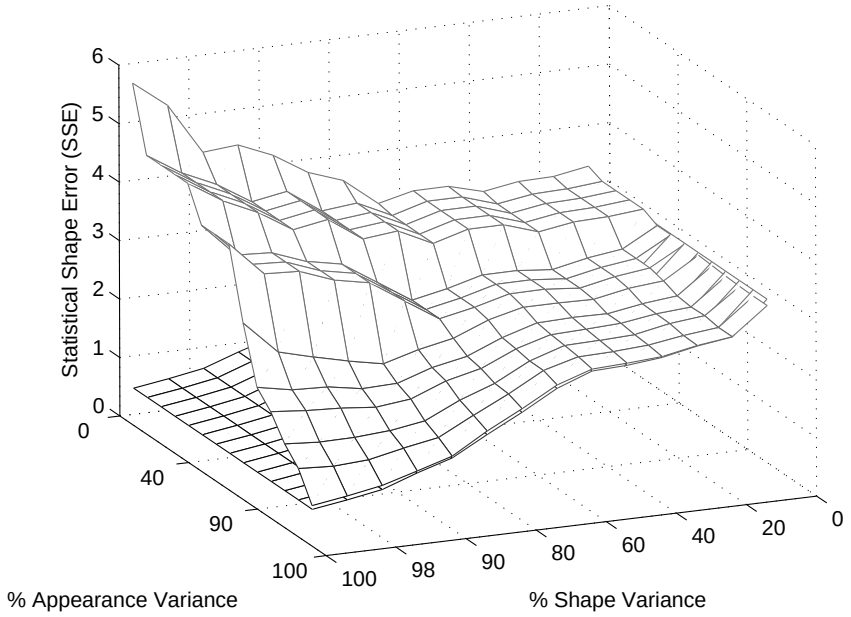


Figure 4.3: *Seen* contexts analysis. The bottom curve represents the initial SSE of equation (4.1) averaged over all the 40 images. The top curve shows the average SSE after the fitting process has ran. Various amounts of shape and appearance variances are tested.

90% shape variance is higher than 60%, the model can bend more when the fitting diverges, but this does not penalize more the frequency of convergence. Since our protocol globally consider all trials fitting accuracy, even the worst cases are considered. Therefore the model should globally be more rigid (less shape variance) not to bend to much on difficult trials and spoil the average SSE result.

In [57] it is not explained why more shape components engender a loss in fitting accuracy (leading to a lower frequency of convergence). We found this can be explained by operating the same test on faces that are known from the AAM: the test faces are also the training faces.

4.2.3 Fitting known faces

In this test, we use all 40 images to train the AAM and one by one, we fit onto these same faces by retaining a variable amount of shape and appearance variance. On figure 4.3 we observe the curve showing the average SSE score on all 40 face images after 60 iterations of the fitting process for each combination of shape and appearance variance tested to build the AAM. It is plot against the bottom curve showing the average SSE score at the starting position of the AAM.

We observe that for full appearance (100% variance retained), the fitting stays onto its best, initial position. The particularity for an AAM when 100% appearance variance is retained, is that it can fully reconstruct the appearance of a seen face. Now, if we observe the curve behavior when some appearance variance is removed from the AAM (the observation holds for any amount of shape variance), we see that the fitting leaves its best, initial position then losing some fitting accuracy.

The only parameter that has changed is the amount of appearance variance that is retained to build the AAM. When less that 100% variance is retained, the appearance statistics cannot fully explain the face appearances: some visual aspects on the input picture face cannot be explained by the AAM

appearance statistics. Since the removal of appearance variance from the model beggets the fitting inaccuracy, we can conclude that the AAM appearance inability to fully express the image is then responsible for the loss of fitting accuracy.

What happens in practice is the following. The difference between input image appearance and model instance of appearance cannot fully be minimized for the best position given to the AAM at initialization. Therefore, the optimization process uses other ways to further reduce the error. It uses the model deformability to bend the model and spread out the residual error as much as it is possible. After all, its role is not to place the model in the most accurate position, but rather to minimize the sum of squared residual of the difference between the input image appearance and the model instance of appearance. Although we would wish the minimum of the cost function 2.40 to be obtained for the shape that places the model into the most accurate position on the face, this can only be the case when the appearance statistics of the model can fully express the face it sees on the input picture.

It is interesting to notice a phenomena that we already observed in the previous chapter at §3.3.1 when we tested the fitting accuracy of an AAM in a person-specific context. The higher the shape variance retained, the less robust the fitting can potentially be because the optimization process can bend the model largely and provide an inconsistent and inaccurate fitting (if it does not totally diverge). This can be observed on figures 4.3 and 4.2 when appearance variance is low and shape variance is high. For the unseen context represented on figure 4.2, this lack of robustness can also happen for high appearance variance retained: when the appearance of the tested face is far from the appearance training data, the model will be likely to diverge on this face.

4.2.4 Discussion

As an observation on the test for seen faces in §4.2.3, we saw that when the model can fully express the image in terms of appearance (the error in intensity between the model and the image are due to the model misplacement and/or non-optimal appearance parameterization). The fitting optimization process uses the error in intensity to iteratively update the model to a position where this error is minimized, and ideally equals zero. It is assumed that the model parameterization that minimizes the error in intensity correctly aligns the model to the face image. In practice, this is what happens when the model explicitly learnt the image it fits (and when the global minimum is reached). This explains the high fitting accuracy obtained in this context.

When the model appearance cannot fully express the face on the test image, the error in intensity due to this lack of expressivity is considered as being due to the model misplacement. The optimization process tunes the model parameters to minimize the residual error though it does not come from a misplacement. In this case, the minimum error usually does not correspond to the best placement of the model vertices. Indeed, the process bends the model in order to spread out the remaining error in intensity as much as it can to minimize the global error. This makes the model drift away from the sought after shape used as its initial position, *i.e.* fitting accuracy is spoiled. The more deformable the model the more the fitting process can bend it to further minimize error in intensity. For very high deformability the model can even diverge. In the same way for a given deformability (fixed amount of shape variance), the less the appearance variance, the less the model can express the test image data and the worse the fitting accuracy. In the case of fitting on seen faces, this happens when appearance variance is not fully retained (less than 100% of the variance is retained). In the case of fitting on unseen faces, a new face always presents visual aspects that are unknown from the model appearance component and the model always drifts away from the best possible position even when appearance is fully retained.

4.3 Reconstructing unseen data with AAMs of different sizes

When it learns face data, an AAM can fully reconstruct these data, but on an unseen face, some visual aspects of the face will not be explainable by the appearance statistics. Why does this happen? This is because an unseen face is likely to occupy some location in the appearance space which is only partially spanned by the appearance basis. The dimensions of the face appearance that are hidden to the appearance basis cannot be expressed, and the unseen face will only partially be explained. As we saw, this lack of expressivity of the appearance space is responsible for the fitting inaccuracy.

But how could we increase the appearance expressivity of an AAM?

Since an unseen face occupies some appearance space dimensions hidden to the current appearance basis, the more natural thing to do is to learn this face, so to obtain a larger appearance basis able to explain these new dimensions! Yes, but the appearance space is very large, and the subspace that spans all the possible face appearances is huge. Gross *et al.* [57] predict that several thousands faces would be needed to train an appearance basis able to express any unseen faces. The resulting appearance basis would obviously be huge too and difficult to handle. Besides being slow, an optimization scheme is prone to reach local minimas in such a high dimensional space. It can also be difficult to gather such a quantity of training data.

Let's have another look at the problem. The unexplainable data can occupy several locations in the appearance space. Apparently the space directions spanned by the appearance basis do not pass by these locations. How can we manage to make them passing by, or closer to these unexplained locations in the appearance space?

The idea we propose here is to divide the appearance space into several smaller spaces, each one corresponding to a different portion of the face. With a given fixed amount of training data, let's B_G be the basis composed of the eigencomponents issued from a PCA computed from the data sampled over the whole face, and of the gain and offset components. Let's now $[B_{L1}, \cdots, B_{Li}, \cdots, B_{Ln}]$ be all the bases encompassing smaller parts of the face. They are trained on the same fixed amount of data, but here sampled according to the face portion they respectively represent. They also have independent gain and offset components.

From such a local appearance basis, we can expect two interesting phenomena to happen.

The first concerns the local gain and offset of one basis B_{Li}. The global B_G offset and gain components will reconstruct the whole face as well as possible, spreading the unreduceable error all over the face. The local B_{Li} that covers a subpart of the surface covered by B_G, will reconstruct only this locality. B_{Li}'s offset and gain components will focus on this subpart to reconstruct and they will generally do it better than the global basis B_G that had to focus on the whole face. In other words, B_G do not pay a particular attention to the locality since it concentrates on the face in its globality, whereas B_{Li} only has to focus on the locality, then it can represent it better. This is the first reason why an AAM covering a smaller part of the face could commit less reconstruction error on this location than does a global model.

The second phenomenom concerns the eigencomponents of one basis B_{Li}. It is true that when a fixed amount of training data is learnt by B_G or by one B_{Li}, and 100% appearance variance is retained to form both basis, the learnt data should be reconstructed identically well by one or the other basis on the face locality they have in common. However, B_G and B_{Li} ability to represent unseen data actually differ, leaving the advantage to the local basis. One could think that the global eigencomponents already contain the information concerning the locality, common to B_G and B_{Li}. This is not exact since the PCA, computed globally or locally, conducts to different directions for the eigencomponents. We actually drop the conditioning between the face's parts. If we chopped B_G's components to make them comparable to B_{Li}'s, B_G's components would not describe the same directions. B_G components will tend to express a face in its globality, with no particular care for localities. One B_{Li} components

will only express a face locality. Their direction in the sub-appearance space should definitely better express the unseen possibilities for this locality than would do B_G's components. They should pass closer to the unexplained data presented by an unseen face, thus making less reconstruction error on this location than a global model. This is the second reason why we decide to experimentaly explore the reconstruction ability of AAMs representing smaller parts of the face.

In the following we first introduce a test for investigating the AAM generability power in function of the training set dimension. In the same fashion we will use the test to compare the generability power of two AAMs trained on the same data, but of different structure and size. We will then introduce the concept of *global model* refering to the whole face model used so far, and *local model* refering to a model describing a small part of the face. This concept is close to the concept of *segmented models* introduced by Blanz and Vetter in [21] which is used for fitting 3D morphable models.

4.3.1 AAM shape and appearance generability

The test we do is inspired from [56]'s test at section 3.2 and we repeat it partly as an introduction for our investigation on model size. We test the shape and appearance generability of an AAM. For this we use our 40 image pool taken from the AR database and labelled several times. We basically test the generative capability of an AAM trained with a varying number N_t of training images: how fine can be the reconstruction of a frontal and neutral face with an AAM built from N_t training faces of different identity? For each N_t, we perform 40 tests, one per image in our image pool. For each image to reconstruct, we build an AAM from N_t images selected randomly among the 39 remaining images (the test image has been discarded from candidate images used for training). In [56], the AAM is built by keeping only 95% of variance for both shape and appearance subspaces following the standard way to build AAMs. We retain 100% variance for shape and appearance since we want to test the maximum generability power of an AAM. The test image is composed of a shape $\boldsymbol{s}$ and an appearance $\boldsymbol{A}$ that we respectively project onto the shape and the appearance subspace of the AAM. To measure the shape error reconstruction we compute the point-to-point error between the original shape and the reconstructed shape. The appearance error reconstruction is computed by averaging the absolute difference between the original test image appearance and its reconstruction after projection onto the appearance subspace: we obtain an average of absolute gray level error for the reconstruction of $\boldsymbol{A}$.

For a given N_t we then compute 40 shape and appearance reconstructions. Their average errors are reported onto the curves on figure 4.4 (a) and (b).

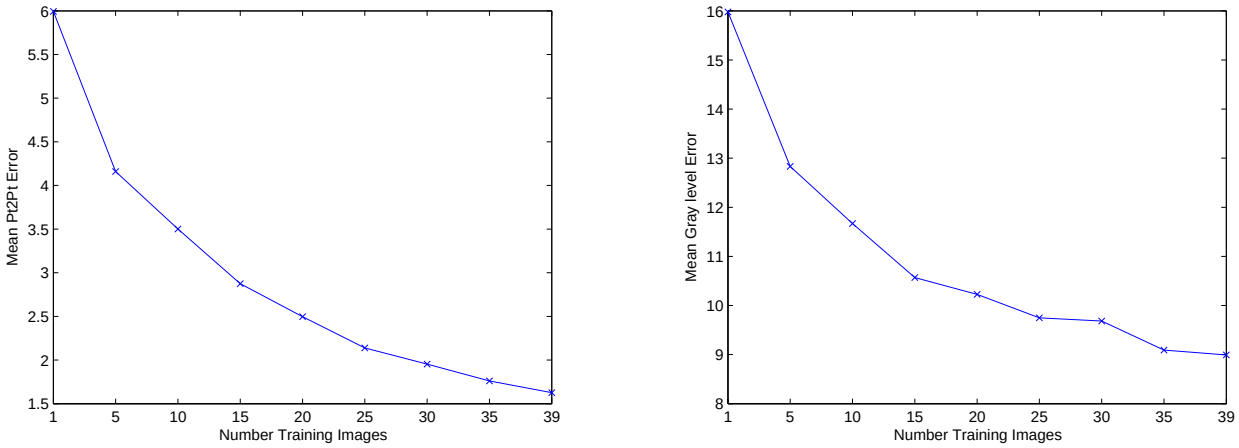
To observe the generability behavior of the AAM, up to 90 training images are used in [56]. Results of appearance reconstruction error are even provided for the test extended to 190 training images to highlight the difficulty for appearance subspace to generalize to any face appearance. In [91, page 20], a similar test and similar result is presented showing results for 200 training data.

In our case, 40 images is few to properly visualise the reconstruction behavior of an AAM for increasing amount of training data and the figure 4.4 represents only partly the results shown in [56, 91]. The reader is invited to consult those works to have a better understanding of how an AAM can generalize when the training set increases.

For what we are concerned, the introduction of the generability test will now allow to compare the performance of AAMs of different size.

4.3.2 Local versus global model: generability comparison

The test is based on the generability test we have introduced. We will use the same AAM as we have used so far, and we now build another smaller AAM, that describes the left eye and left eyebrow of the training faces. We will talk about *global model* for the AAM that covers the whole face, and *local model* for the smaller AAM. Figure 4.5 presents a training face with the local model in (a) and the



(a) shape generability (b) appearance generability

Figure 4.4: Shape and appearance average reconstruction error for different amount of training images.

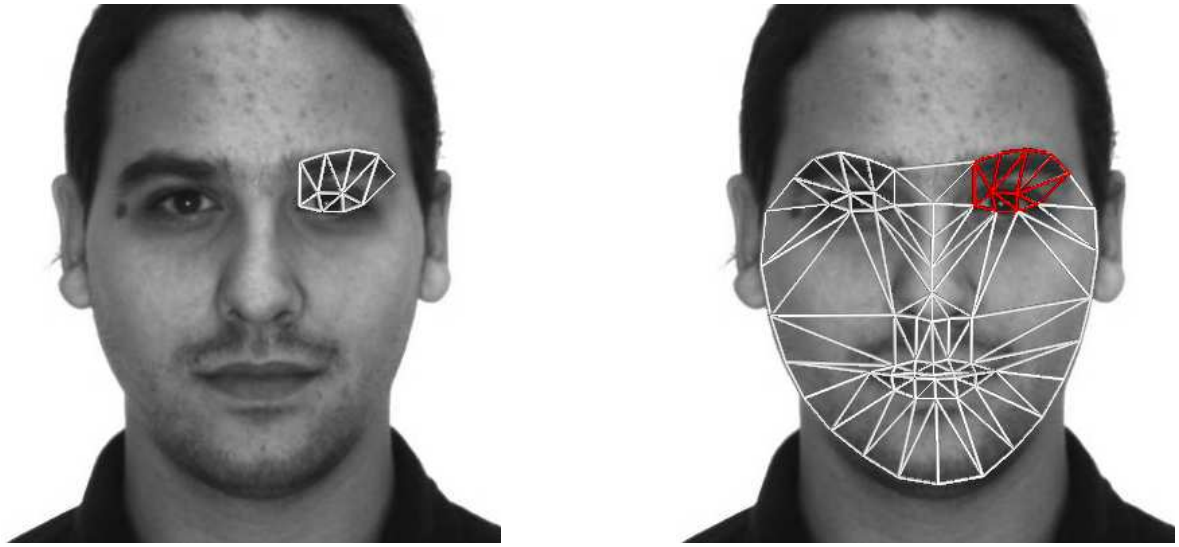
global model in (b). The test consists to build both AAMs from the same training images, and to reconstruct a test image different from training images with both models. Of course the area covered by the global model is larger than the area covered by the local model. We compare both model reconstruction performances on the area they have in common. The global model point-to-point error is then computed only for vertices that are common to those of the local model, and its mean gray-level absolute error is computed only over the pixels that concern the area also covered by the local model, i.e., the left eye, the left eyebrow, and the part located in between.

In the same fashion as before, we build global and local AAMs from N_t training images that are identical for both AAMs and that are different from the test image. For various dimensions N_t of the training set we process 40 reconstruction tests and compute the average of the 40 shape and appearance error reconstructions to report the result on curves presented figure 4.6. For the global model (as well as for the local model of course) the reconstruction error is evaluated for shape on vertices that are common to the local model only, and for appearance on the pixels that lies under the local model only.

We observe that the reconstruction of unknown shape and appearance data is better for the local model: the generability power is then higher for an AAM of smaller dimension (less vertices, less *usefull pixels* to model).

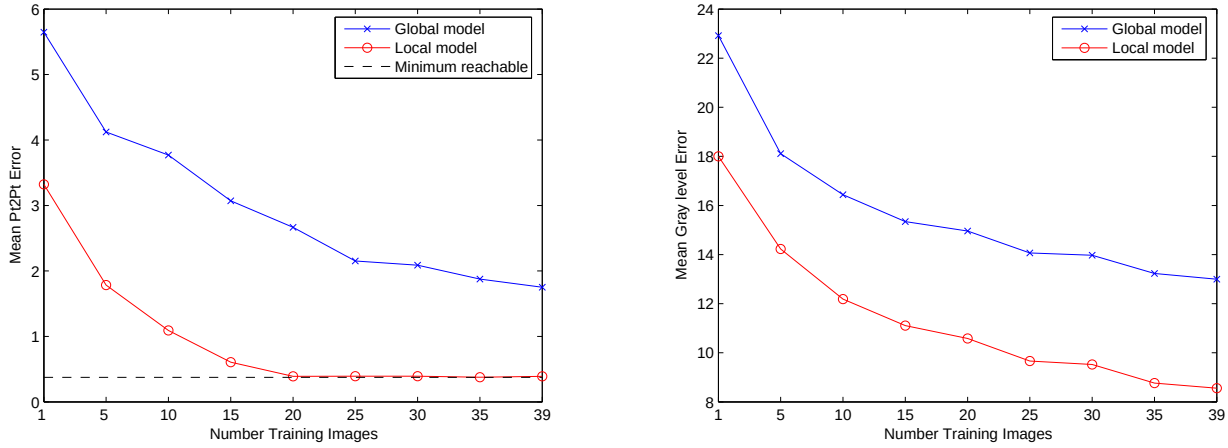
On figure 4.6 (a), for N_t equals 20 or more the shape reconstruction of the local model achieves the minimum error possible (defined by the average point-to-point distances between the ground-truth shapes and the same shape which vertices coordinates are rounded to closest integer value). This is due to the small dimensionality of the shape space for the local AAM. It is composed of 11 vertices, equivalent to 22 dimensions. With as few as 18 shape components (available for N_t equals 19 or more) the whole shape space is spanned by these vectors combined to the 4 components dedicated to similarity transformations. Any point in this space is fully described by shape components and is therefore reconstructed equivalently to itself when projecting it onto these vectorial components that span the whole space. Then the ground-truth shape is reconstructed perfectly up to the rounding to next integer pixels.

Note that the potentiality for shape space to build the perfect shape of a given test face does not mean that a fitting process could easily fit the AAM on the perfect position: we saw that when an



(a) (b)

Figure 4.5: Illustration of a training face with the local model in (a) and the global model in (b) where both models are placed onto the ground-truth position. The part of the global model which is common with the local model has been evidentiated in red.



(a) shape generability (b) appearance generability

Figure 4.6: Shape and appearance average reconstruction error for different amount of training images. We visualize reconstruction performance for both local and global model over their common area and vertices.

AAM is built at full shape variance, it is also poorly robust and prone to diverge. Here, we only test the generability performance of a local AAM. The fitting performances will be tested later in this

chapter.

4.4 Local Models

The better generability performance of smaller models leads us to the idea of *segmented models* or *local models*. To fit with more accuracy each point of interest, a higher appearance expressivity should improve the fitting quality. Each local model we create will gather a certain number of points of interest of the global model used so far. We build a collection of local models to focus on each facial feature.

Since a local model is an AAM, it is composed of a shape defined by a configuration of points and an appearance defined by the area contained within the triangulation operated on these points. The fitting of AAMs onto a picture is possible thanks to visual gradients of both picture and appearance statistics of the AAM. Assuming that they are common to both the picture and the AAM statistics, the gradients are put into correspondence by optimization. Usually, a point of interest is chosen on the face for its peculiarity and its semantical definition, what means in general that this point and its surrounding are rich in visual gradients to which the experts attribute a semantical meaning. Thus, a triangulation performed only on the selected points of interest for a local model conduct to obtain an incompletely defined appearance of the model: the appearance of points of interest that bound the external edge of the model are defined only partially. We then need to add a layer of *support points* around the external points of interest. In this way, all points of interest can visually be fully described and the gradients that define them can be seen at 360 degrees with this *supported* local model.

The support points must be added to the training images. The larger the support, the more area is covered by the model onto the picture, what gives more robustness when fitting the model that *has more visibility of the image gradients*. But the smaller the support, the less sensitive to unexplained visual data and the more accurate can be the model.

Then we choose to set up a coarse to fine approach starting from large, robust and not highly accurate models, to small and accurate models that need to be very well initialized to converge into the right minima.

Each local model is built from a bunch of face landmarks that describe the shape of one or more face features. The choice of the landmarks composing each model was done arbitrarily. However, several attempts were done on different kind of models to finally adopt a solution that seemed more satisfactory to us.

Figure 4.7 presents a schematic view of the models we use and on how they should be fitted in turn onto the face.

Hereafter is presented the way to build the set of models and to fit them in a three steps coarse-to-fine strategy.

4.4.1 Building Local Models

To define what could be the optimal support area of each local model is a complex task that we did not investigate. In a first time we simply tried to place support points manually on the training pictures at an arbitrary distance of external points of interest, what globally gave very good results but sometimes led to observe fitting divergence due the shape variability of the support points: since they are generally not defined on image gradients and they can be *pushed* by local unexplained pixels, these points can drift away on the picture and sometimes bring the points of interest to diverge in the same movement.

To address the problem of divergence we manage to place these support points on training images in such a way that shape analysis and PCA on local models engender a null tendency to deform on these

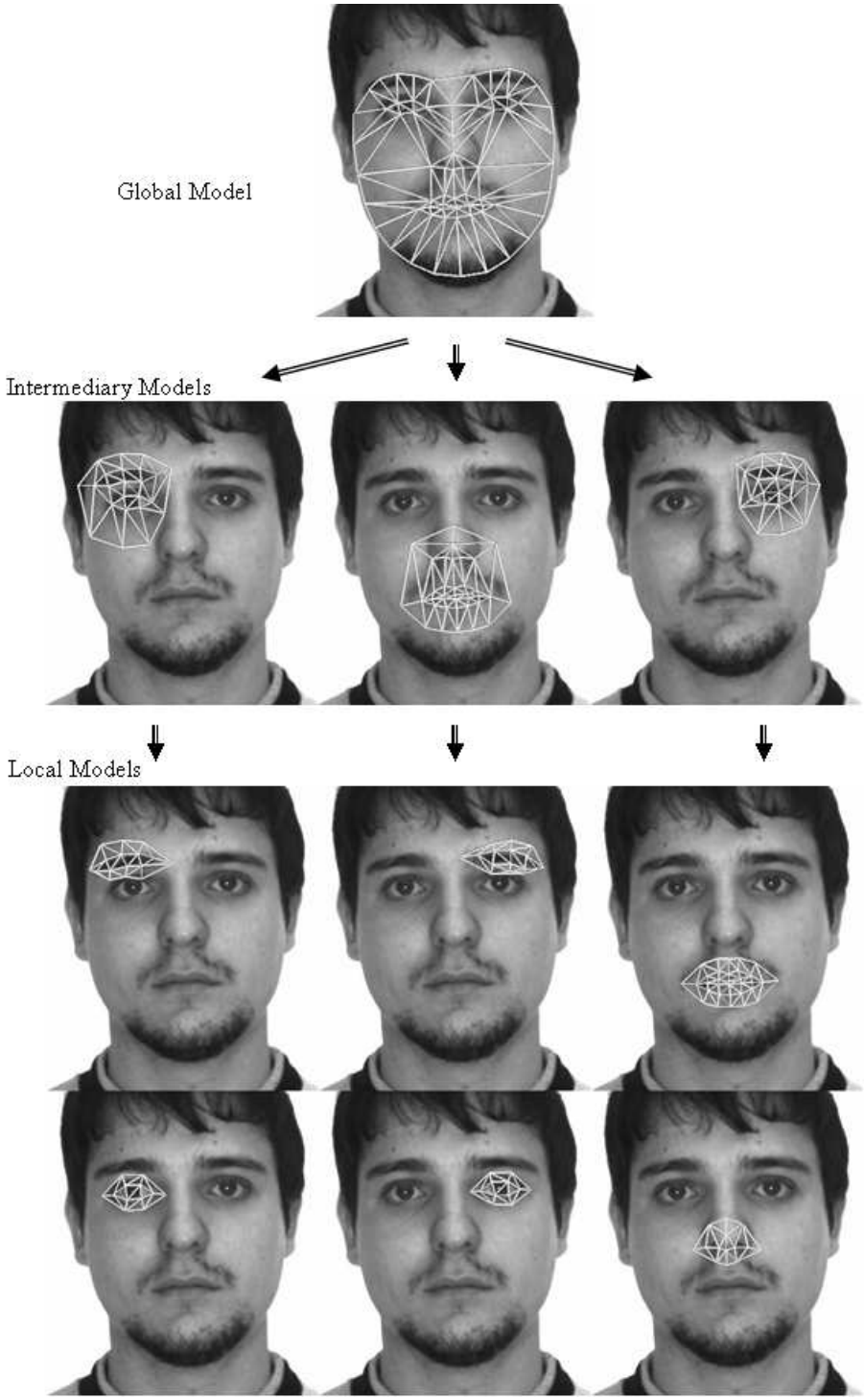


Figure 4.7: Illustration of the models used to fit the face. A global model is fit first and gives a first fitting result. From this initialization a set of intermediary models are ran to further refine the fitting accuracy. Eventually, the local models dedicated to each facial feature are ran to fit these features more accurately.

points: these models are consequently more stable and better prone to converge than models where support points are placed manually on training images. This support points insertion onto training images is done semi-automatically, what is another concrete advantage. Only few parameters must be fixed manually as we will see in the following.

To compute support point location we work in the pre-shape space. Let I_{tr} be the training image number $tr \subset [1, ..., N]$ where N is the number of training images, and $Cinterest_{tr}$ be the configuration of landmarks of interest (landmarks concerned by the local model we want to build) in this training image I_{tr}. A shape normalization is applied on the $Cinterest_{tr}$ for all $tr \subset [1, ..., N]$ to obtain a collection of centered and normalized shapes S_{tr} consequently adjusted in rotation to minimize their distance point to point (see §2.1.1 for reference). The mean shape S_{mean} can thus be computed from all S_{tr} that gravitate around it. As an illustration we follow the construction of the local model used for the mouth. Figure 4.8 presents the superimposed S_{tr} shapes and the mean shape relative to mouth landmarks.

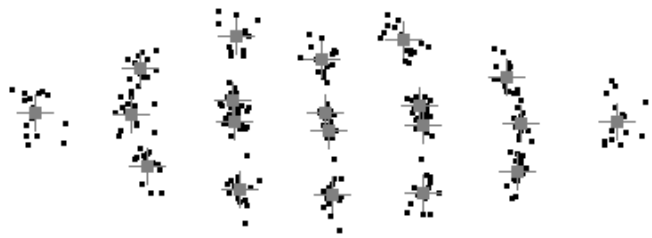


Figure 4.8: Illustration of the normalized mouth training configurations. Superposed normalized configurations are represented with white plots and the mean shape with gray crosses.

As shown on figure 4.9 for the special example of mouth local model construction, let Lk_i be the vertices that compose S_{mean} and $ExtLk_i$ the most external points among the Lk_i (the easiest way consists to pick up those points manually).

The $ExtLk_i$ are ordered in such a way that they describe a circulation along the mesh perimeter. The sense of the circulation (two possibilities) defines the side where the support-points Spt_i will be placed. Let's say that in the circulation, $ExtLk_{i-1}$ is located before $ExtLk_i$ which is itself placed before $ExtLk_{i+1}$. If n_{Ext} is the number of external points, i can assume any value, and will correspond to $mod(i, n_{Ext})$ such that vertex $ExtLk_{-1}$ will also be $ExtLk_{n_{Ext}}$ and $ExtLk_{n_{Ext}+1}$ will be $ExtLk_1$.

For each $ExtLk_i$ we define one Spt_i which is placed according to $ExtLk_{i-1}$, $ExtLk_i$ and $ExtLk_{i+1}$. The bissectrice of the angle $ExtLk_{i-1}, \widehat{ExtLk_i}, ExtLk_{i+1}$ is computed and Spt_i is placed onto it at an arbitrary distance on the side external to the mesh formed from the $ExtLk_i$. We will consequently obtain one Spt_i per $ExtLk_i$. Figure 4.10 illustrates Spt_i construction process.

The Spt_i are computed to surround the mesh processed on S_{mean} vertices. They also surround each S_{tr} since they gravitate around S_{mean}. What we do as illustrated on figure 4.11 is in turn to cling the support-points to each S_{tr} to form a unique shape $Scomplete_{tr}$ that we can reproject onto the training image number tr: $Scomplete_{tr}$ is adjusted in translation, scale and rotation to form C_{tr} whose Lk_i vertices perfectly correspond to $Cinterest_{tr}$. C_{tr} then combines $Cinterest_{tr}$ with newly obtained $Csupport_{tr}$ configuration of support-points. The reprojection of the *supported shape* is illustrated on figure 4.12.

We did not say how we choose the arbitrary distance separating $ExtLk_i$ and Spt_i. It is adjusted arbitrarily by simply verifying the visual aspect of configurations C_{tr} onto each training image I_{tr}. If the configurations look nicely designed onto the training faces, we retain the set of distances chosen to separate the $ExtLk_i$ from Spt_i, otherwise we try some new distances to suit our convenience. This

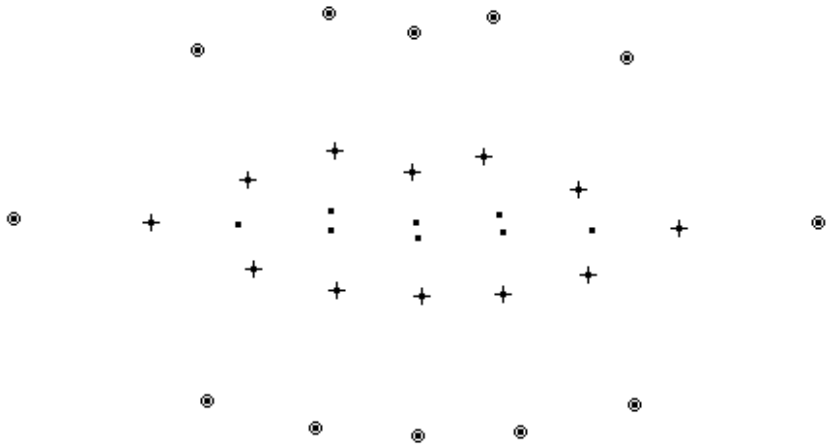


Figure 4.9: Illustration of the mean shape and of the support points. The mean shape is composed of landmarks Lk_i among which we find the more external ones $ExtLk_i$ represented with crosses and the support points Spt_i placed around the mean shape are represented with plotted circles.

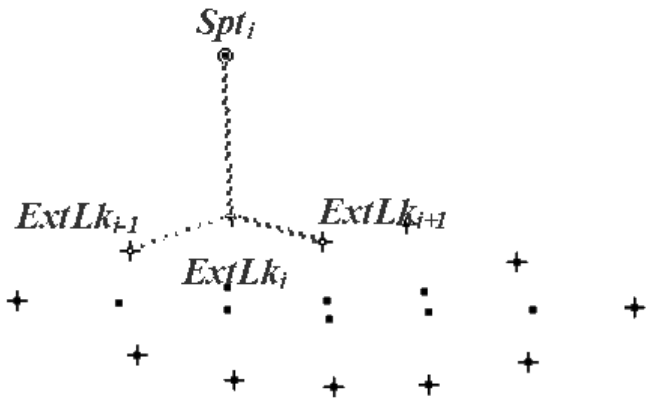


Figure 4.10: Along the circulation of the external mean shape vertices, we select one $ExtLk_i$ and its neighbors $ExtLk_{i-1}$ and $ExtLk_{i+1}$ to compute the location of the support point Spt_i associated to $ExtLk_i$. Spt_i is placed on the bissectrice line of the angle $ExtLk_{i-1}, \widehat{ExtLk_i}, ExtLk_{i+1}$. The process is repeated for each $ExtLk_i$.

choice is purely subjective but the definition of an optimal set of distances depends on too many parameters like the nature and quantity of training images (the optimization task should be repeated any time we change the training set of images), and a cost function is also difficult to design or would require huge complexity of fitting performance evaluation for each new setting of the distances to be optimized.

Some works have focussed on the possibility to automatically generate AAM training data: from a set of unlabelled or badly labelled training images, an (usually very costly) optimization is processed to automatically label the images and build a well performing AAM. The theory of such a task is

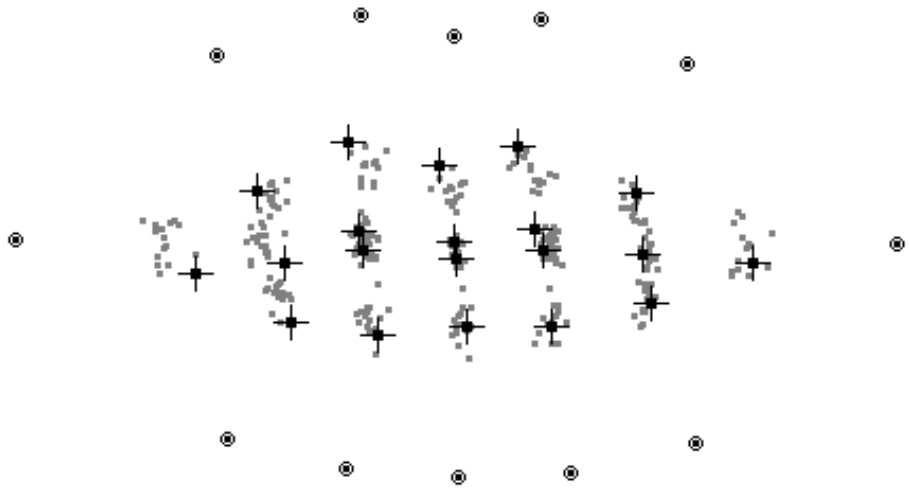


Figure 4.11: Support points Spt_i represented with dotted circles surround all normalized training shapes S_{tr} represented with gray plots. One of the S_{tr} (here S_1) however is evidentiated with black crosses. Support points are clung to this shape to form a new configuration that we reproject onto the training image I_{tr}.

Figure 4.12: Training image on which the supported shape has been reprojected.

well explained in [10] and a practical result is provided for AAM construction to generate very simple patterns which are far from face complexity. De La Torre in [65] makes use of genetic algorithms to perform a complex but smart automatic AAM generation on new images of a person of whom we already labelled images. This work might inspire a solution to define the distances $ExtLk_i$ to Spt_i automatically.

In this way we re-label each training images by adding the layer of support-points. The aim of the operation is to obtain a set of training configurations C_{tr} that produce a null variability of shape onto the support-points (all corresponding support points along training images merge onto the same spatial location during the shape analysis). The fitting with such a model results more robust than when variability is allowed on support-points as well as points of interest.

Figure 4.13: The shape analysis operated on all C_{tr} lead to support-points that are subject to shape variability.

The null shape variability should then be ensured by the fact that we precisely choose the support-points to belong to the same location in the shape space (*i.e.*, no shape variability). In reality, the shape analysis performed onto the newly built C_{tr} configurations does not lead to the same repositionning in the shape space as we illustrate it on figure 4.13. We remind that each shape S_{tr} were obtained from $Cinterest_{tr}$ was normalized to unity, and the addition of the support-points broke this unity norm: the support-points were aligned for each S_{tr}, but the resulting shape vector combining S_{tr} and the support-points was not normalized anymore. Normalization brings the support-points to dissociate in the shape space and to lose the shape invariability property. What we do is to iteratively recompute the support-points to be the mean of the support-points obtained after the novel normalization. After a couple of iterations (typically 3 or 4), the re-normalized support-points converge onto the same spacial location, this time really ensuring the shape invariance on those points as illustrated on figure 4.14.

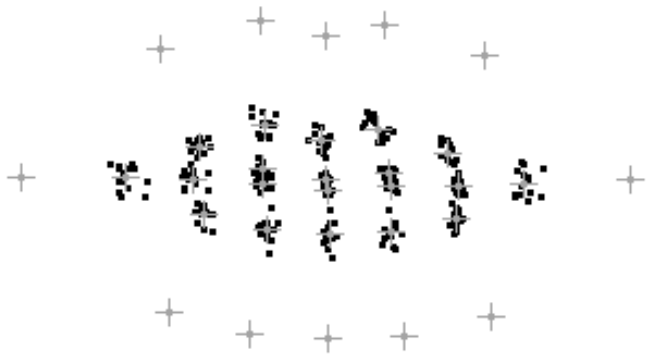
When we use the model to fit onto a picture, we effectively observe that the support-points of the model do not deform, the structure they form is only subject to similarity transformations (translations, scale and rotation). Only the points of interest of the model can deform in combination to the similarity transformations common to all points of the model.

4.4.2 Three steps coarse-to-fine fitting strategy

In previous subsection we saw how to build the training images with landmarks of interest and their support-points. Illustrations were given for the mouth local model construction, but the same process is operated for each feature or group of features that we want to represent by local models. The idea is to fit in turn the usually more robust larger models that present a better initialization location for smaller models that are less robust to initial perturbation but are potentially more accurate than larger models.

First tests led us to choose a 3 stage coarse-to-fine process: a two stage approach seemed too few to reach robustness and accuracy and a 3 stage one is enough to focus on each single feature with good accuracy due to the first two stages. First a global model is ran on the face, initialized with a face and

(a) First shape analysis and recomputation of the support-points.



(b) Fourth shape analysis and recomputation of the support-points.

Figure 4.14: To suppress the shape variability on the support-points, we iteratively repeat the following process: all C_{tr} are projected into the shape normalized space, the mean-shape is computed, and the mean support-points are clung onto each S_{tr} to form a new C_{tr} that we reproject onto the training image. On (a), the first iteration is shown with all C_{tr} represented with black dots and the mean shape represented with gray crosses. On (b), the same at the fourth iteration. After four iterations of the process, we obtain a perfect shape invariability of the support-points that all correspond to the mean support-points.

eye detector. The converged position of the global model is used to run three *semi-local* models that gather two features each (They are presented in figure 4.7). Their fitted position is used to initialize the six smallest models, one for each feature of the face.

4.4.2.1 Initializing the global model

As first initialization we use a face and eye detector proposed by Fasel et al. in [48] available on-line. It is inspired of the Viola & Jones boosted Haar-feature based classifier presented in [106]. With high rate of detection, the detector provides a square bounding box including the whole face and a location for the center of each eye of the person. The test of the detector on the 133 different neutral face images of the AR database worked perfectly for face detection (all face features were found inside of

the bounding box whose area includes face almost uniquely) and failed only 3 times during the eye detection: only one eye was detected on the 3 failed cases.

The center of the eyes is not always accurately detected. However the rigid global model happens to be able to deal with most of bad initializations. Let's see how we initialize the model position.

The global model we use is made of vertices corresponding to a particular location on the face. If we know the position of only two of these points on the face presents on the input picture, we can place the whole model (its mean shape) onto the picture: when we constrain two model vertices to assume a particular location, this also define the location of all other vertices that are linked to the first two by fixed geometrical properties intrinsically defined on the mean shape model (obviously the model cannot deform and is only subject to similarity transformations). But the eye detector is supposed to detect eye center position, point that does not correspond to any vertex on the model we use. However on the model we use the inner (called medial canthus) and outer corners of the eyes. On the 40 face images we labeled manually, we checked the distance ratio between both eye center distance and between external eye corner distance. It happens to be equal to 1.42 in average, and the standard deviation on all computed ratios is pretty small, equals to 0.0258. We choose to use the constant 1.42 to find the position of the external eye corner points when we are given both eye centers and we make some other assumptions: if C_r and C_l are the right and left eye centers, and E_r and E_l are the right and left external eye corners, the segment $\overline{E_rE_l}$ is 1.42 times longer than $\overline{C_rC_l}$ and these segments are colinear and have the same middle point. Provided the center of both eyes, we can obtain an estimate of the eye external corners and we find the initial position of the model. An illustration of a typical result is shown on figure 4.15. Both eye centers might not be found perfectly as can be seen.

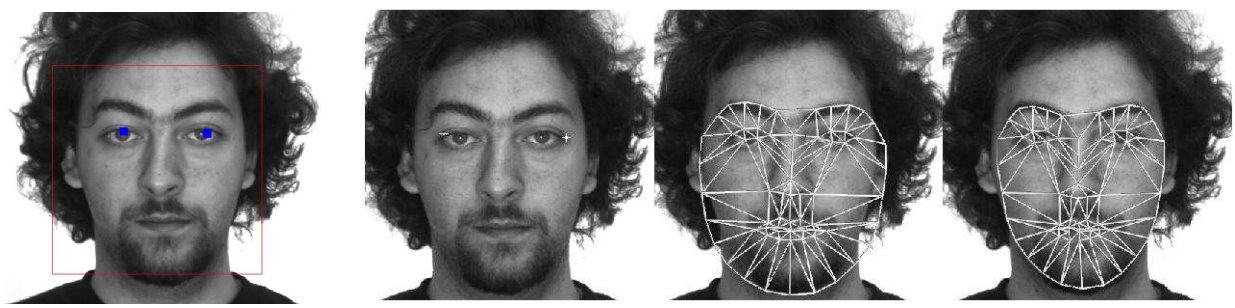


Figure 4.15: Typical face and eye-center detection. The eye external corner are interpolated from the distance and direction of the two eye-centers. The global model is initialized by placing into correspondance the model eye corners with those found on the picture. After few iterations, the model converges to a new better location: this is an initialization for the intermediary models.

On figure 4.16 we present another likely situation in which the eye detection was not as good.

On figure 4.17 is illustrated the only one trial for which an inaccurate eye detection leads to initialize the global model in a location that brings it into a wrong local minima. Considered the rareness of the situation, we did not consistently investigate a solution to detect whether the convergence is correct or not ("shall we keep or reject the fitting?"would be a nice problem to investigate). In the future, it could be interesting to improve the overall system robustness. For now the priority is still to find the way to fit onto new faces, frontal or not, expressive or not.

Another problem that will require more investigation is the iterative process stop criteria. After the unanswered question "is the convergence correct?", another relevant question is "when can we consider that the fitting has converged?"

The classical criteria "stop when the model movement is lower than a certain quantity"is not

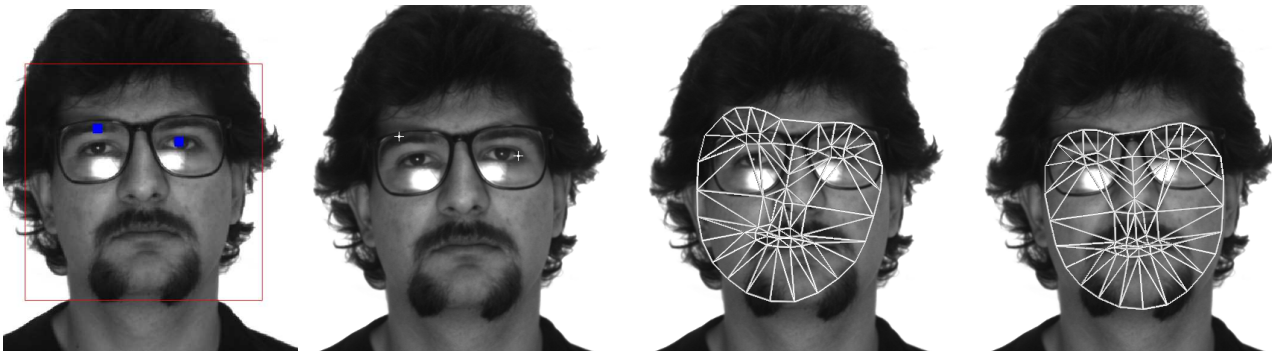


Figure 4.16: In this example, the eye detection seriously lacks accuracy for the right eye center detection. The global model initialization is consequently pretty bad. However we see that the robustness of this rigid model can manage with this situation and leads the model to converge to the right position.

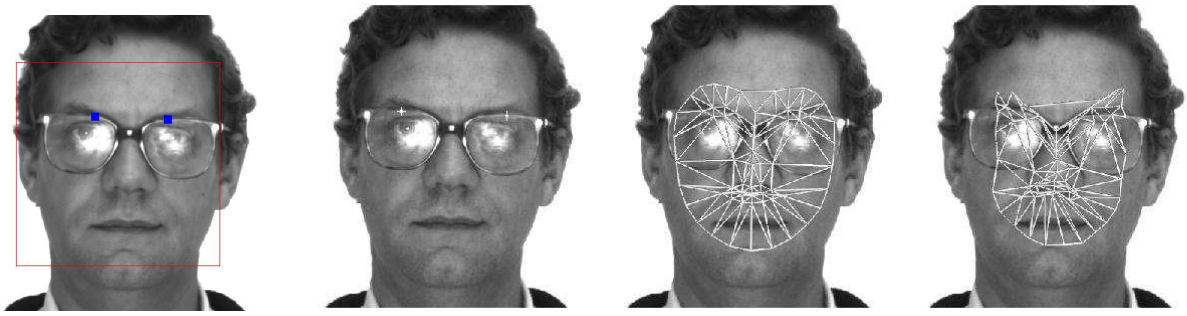


Figure 4.17: Among the 130 faces where eyes were correctly detected, this example is the only one presenting an eye detection bad enough to make the global model fail at finding the right minima.

satisfactory since this quantity should be predicted for each situation. Tests with fixed quantity have not provided interesting results: the process is often stopped before it should, or at the contrary, it stops too long after it has converged already. We tried some tricks that consider the cost function value at each iteration. We chose a simple criteria: "stop when the cost function does not decrease from an iteration to the following". This globally works well, but it sometimes happens to make the system stop too early. It is important to note from the literature that the definition of the most adapted stopping criteria is a the object of complicated optimization problems.

For the presented tests, we launch the fitting process for a certain number of iterations, depending on the model we fit, and for which we are sure the convergence is reached if it can be. This is actually the most reliable criteria (if it is one) we found. For example the rigid global model is launched for 30 iterations, but usually need from 2 to 10 iterations to reach the convergence. Thirty iterations running on Matlab are typically performed in 3.5s on a pentium centrino 1.7GHz.

Please note that initialization of the global model could be improved in different manner: for example, if a mouth detector is available as well as the eye detector, three points would be available that allow to better constrain the spatial global model location onto the input picture. Even though the eyes and mouth were not found in their exact center, the model could be placed in a way that its eyes and mouth can partly see the eye and mouth on the picture, what usually lead to convergence.

When only the eyes are available, an inaccuracy on their center detection can lead to a large error positionning of the mouth that is deduced by geometrical properties. However, the global model built with high appearance variance and 60% shape variance shows to be pretty robust at fitting unseen faces.

4.4.2.2 Initializing intermediary models and local models

Global model position is used to initialize each intermediary model: we keep the vertices the global model has in common with an intermediary model, and we find the intermediary model instance that best matches those vertices, as follows.

Let $\boldsymbol{B}_{succ}$ be the shape generating matrix of one intermediary model: the columns of $\boldsymbol{B}_{succ}$ are the n_C long deformation vectors s_i plus the four similarity transform vectors. Vector $\boldsymbol{s}_{curr}$ represents the vertex coordinates of the global model that are in common with the intermediary model. To this vector we add extra null coordinates for intermediary model vertices that are not in common with global model. $\boldsymbol{s}_{curr}$ thus becomes n_C long. We sort the coordinates in $\boldsymbol{s}_{curr}$ in a way such that they correspond to vertex coordinates defined in the vectors s_i. The instance of intermediary model that best matches its common vertices to those of the global model is found by solving the following optimization problem:

$$\arg\min_{\boldsymbol{p}} \sum_{c=1}^{n_C} \boldsymbol{Q}(c)\left(\boldsymbol{s}_{curr}(c) - \boldsymbol{B}^c_{succ}\boldsymbol{p}\right)^2, \tag{4.2}$$

where $\boldsymbol{B}^c_{succ}$ is the cth row of matrix $\boldsymbol{B}_{succ}$. $\boldsymbol{Q}$ is an n_C long vector of weights set to one for the coordinates of vertices that are common between the models, and to zero for the others. A closed form solution can be computed to find the optimal $\boldsymbol{p}^\dagger$ (computation details are present in annexe 7.4):

$$\boldsymbol{p}^\dagger = (\boldsymbol{K}^T\boldsymbol{K})^{-1}\boldsymbol{K}^T\boldsymbol{B}^T_{succ}diag(\boldsymbol{Q})\boldsymbol{s}_{curr}, \tag{4.3}$$

where $\boldsymbol{K} = \boldsymbol{B}^T_{succ}diag(\boldsymbol{Q})\boldsymbol{B}_{succ}$ and $diag(\boldsymbol{Q})$ is a diagonal matrix, null everywhere excepted on diagonal where the $\boldsymbol{Q}$ vector coefficients are represented. The result $\boldsymbol{p}^\dagger$ of this minimization can be used to instantiate the shape of the intermediary model: $\boldsymbol{s}_{succ} = B_{succ}\boldsymbol{p}^\dagger$.
The process is applied to initialize all intermediary models that are then fitted to the image. Following the same strategy, we use (converged) intermediary model vertices to initialize local models that are in turn fitted to the image.

4.4.3 Evaluation and results

To evaluate the method we perform a leave-one-out test on the 40 face images of the AR database on which we can use the SSE to measure the fitting quality as explained in 4.2.1. The test consists in training all global, intermediary and local models on 39 face images, and to fit onto the remaining picture. We can perform 40 fitting tests onto which we score the fitting error with the SSE measurement.

For the global model used in the first fitting stage, we retain 60% shape variance which shows to be a good trade off between fitting accuracy on unseen faces and robustness to initial perturbations. In the second and third stage fitting, we retain a certain amount of shape components for intermediary and local models. We will determine which amount of shape components it seems more relevant to retain. In all cases, we retain 95% of appearance total variance of the training set.

In 4.4.3.1 we will find which amount of shape components optimizes the fitting quality on unseen faces when using intermediary models. In 4.4.3.2 we will find which amount of shape components optimizes the fitting quality on unseen faces when using local models.

4.4.3.1 Determining the best amount of shape components for intermediary models

As we said, we train the models on 39 face images among 40, and fit onto the remaining face. The global model is trained to retain 60% shape variance, and 95% appearance variance of the training set. From the position provided by this rigid global model, we run the intermediary models in the fashion explained in 4.4.2.2. These intermediary models are trained with 95% appearance variance, and a variable amount of shape components. The test consists in observing the fitting accuracy and robustness according to the number of shape components retained.

On figure 4.18 are presented the 40 fitting results for a number of shape components varying from 0 to 15. The results are presented under a boxes of whisker plot form. Seing those results, we rapidly localise the more advantageous amount of shape components that optimize the results in terms of accuracy, but also of robustness. The criteria we use can be the median error fitting which is a robust measure to potential fitting quality the model generally reaches, the (non-outlier) data deviation around this median point that tells about the robustness we can expect from the fitting to generally result close from the median quality fitting, and the percentage of outliers. Concerning outliers, it is important to identify their reason for occuring in order to be able to predict in what condition of work they should be avoided.

We watched the results shown on figure 4.18 together as the visual fitting results to observe why outliers occur. For eye and eyebrow models, fitting score resulting as outliers come from particular non-learnt specularities on glasses as illustrates figure 4.19. In the pictures we use from AR database, faces are illuminated frontally, what sometimes causes important reflection on glasses for people who wear them. The reflection is usually high and variable according to the glasses' structure and the small variation in head pose. For nose and mouth model, outliers occur on faces presenting particularities that were not learnt from training images, like a thick beard, a mouth not displaying a completely neutral state or an atypical mustach. Figure 4.20 visually illustrates the outliers, *i.e.* , the faces where fitting quality is poor.

In a general manner, any test data far from the training set's pictures may affect negatively the fitting quality.

The choice of the best amount of components is not obvious. For the eye and eyebrow models, overall accuracy gets better when increases the amount of shape components, but it is at the detriment of robustness, and diverging tests will diverge even more. One could claim that diverging fitting should be detected and rejected. In the other hand a perfect accuracy is not required at the intermediary fitting stage since we rely on the local models to provide it.

Instead of relying on results plotted on figure 4.18, we visually inspect the 40 fitting results one by one, for each amount of shape components. For 10 shape components retained, good fitting are very good indeed, but at the same time incorrect fittings are very bad what reduces hopes for local models to get any improvement from such initialization. In the other hand, for a shape component amount of 4, the global results are less accurate but far good enough for an intermediary fitting stage, and incorrect fittings are often good enough for initialization of local models. We then decide to retain a reasonnable amount of shape components, fostering a good fitting robustness and a fitting accuracy correct enough for initializing the local models properly. We decide to retain 4 shape components for the eye and eyebrow models. This corresponds to 80% of the training set total shape variance. For the mouth and nose model, the result clearly indicates that our choice must be done between 2 and 7 shape components. Fitting accuracy is clearly better for this range of components and robustness seems correct. For less components, accuracy is sacrificed for robustness: we deal with a too rigid model. For more components, the model is too flexible and bends too much in presence of unexplained visual details on the unseen face. Once again, our choice is made on 4 shape components, equivalent to 75% of the training set total shape variance.

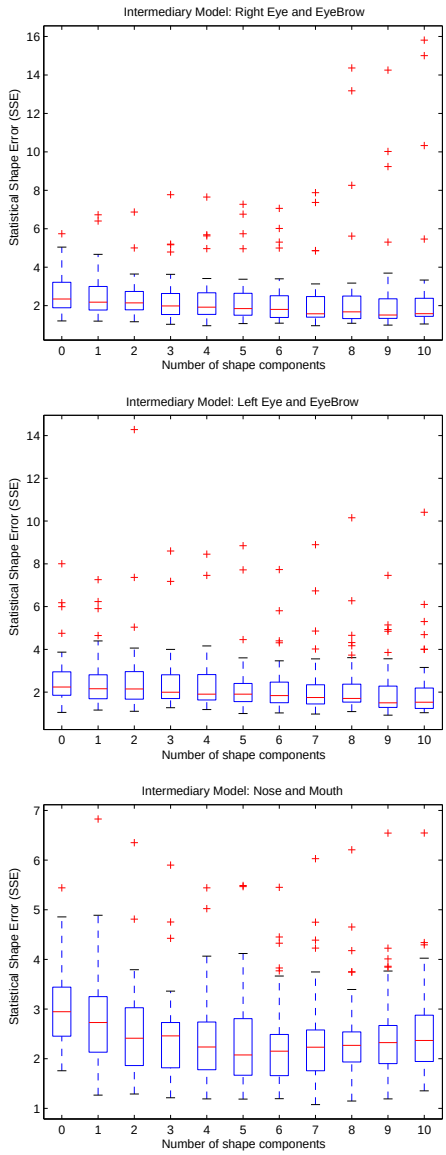


Figure 4.18: Intermediary Model fitting results in function of the number of shape components retained. Results are presented under boxes of whisker plot form, where median value among the 40 fitting error values is represented with a red tick inside of a box bounded by two lines representing the lower and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers and are represented by red crosses.

We observe for 4 shape components retained that most of fitting results are very accurate. On figure 4.21 we show some typical fitting results we obtain with global, intermediary, or local models.

4.4.3.2 Determining the best amount of shape components for local models

We then set the amount of shape components to gather 75% of the shape total variance of the training set (here keeping 4 shape components) for the nose and mouth intermediary model, and to gather

Figure 4.19: Bad fittings of intermediary model leading to outliers in fitting score for eye and eyebrow intermediary model. They are due to non-learnt glasses specularity.

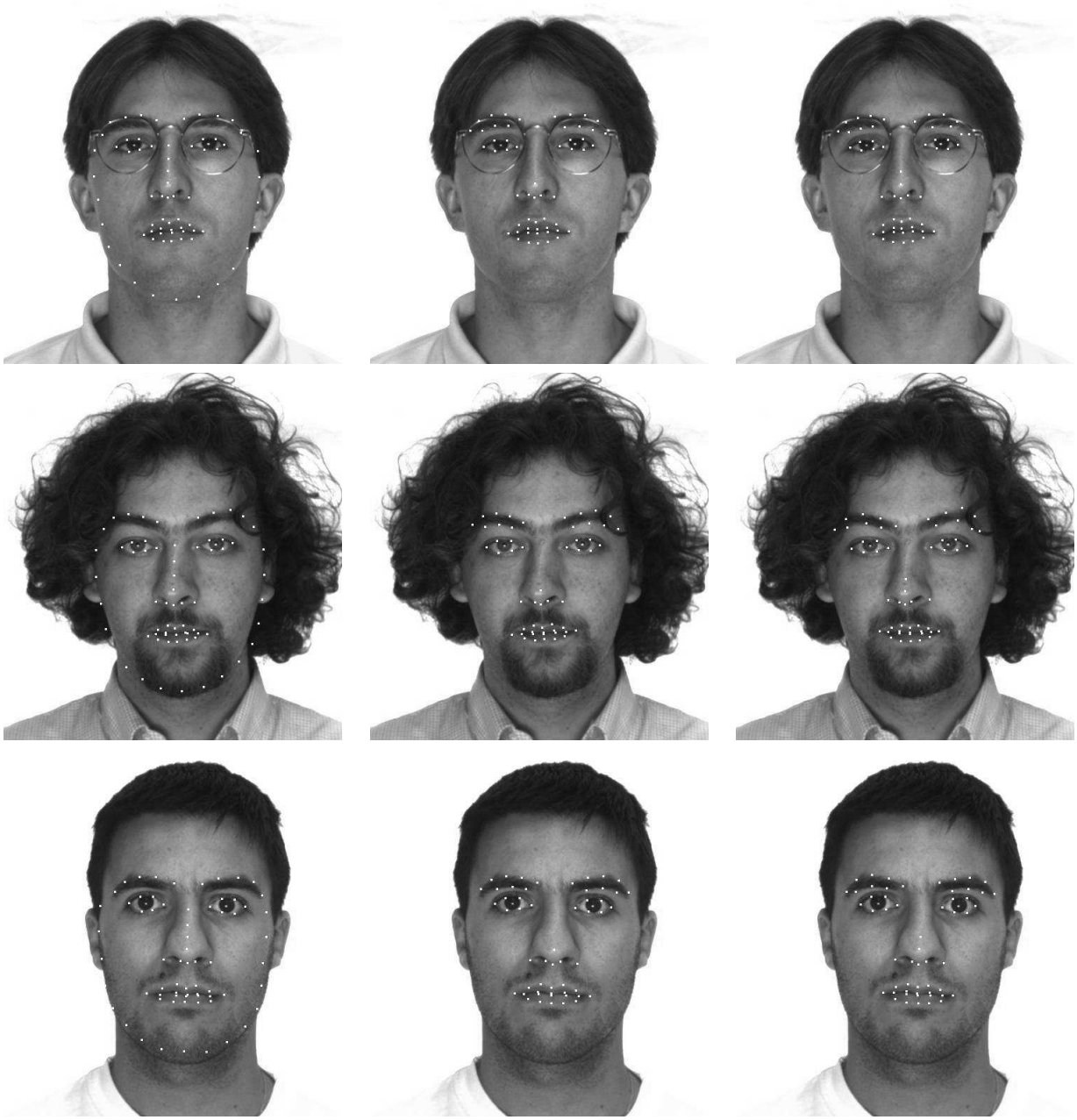
Figure 4.20: Bad fittings of intermediary model leading to outliers in fitting score for nose and mouth intermediary model. They are due to non-learnt particularities on or around the mouth: a non-learnt mustach, a non-learnt thick beard or a not completely closed mouth, which is an unknown pattern.

80% shape variance (keeping 4 components) for each eye and eyebrow intermediary model, and we run the local models from the fitting position provided by the intermediary models.

In the same way as for intermediary models, we repeat the fitting test for different amount of shape components retained for each local model. We recall that here too, the fit face is unknown to the AAMs: the training process of all models is repeated for each face, in order to exclude it from the 40 face picture training set.

The results are plot on figure 4.22 in the same fashion as for intermediary models. Interpretation of best shape component amount is not obvious for all models. For mouth and nose, the shape component amounts can be done rapidly on three components for mouth local model (what represents 70% of total shape variance), and on five components for nose local model (what represents 94% of shape variance).

Eye and eyebrow model results are somehow more difficult to interpretate. Outliers are represented and are useful to evaluate the fitting robustness, but they spoil the visibility of the results. We then re-plot those boxes on figure 4.23 where fitting errors are limited to a threshold value that makes



Global Intermediary Local

Figure 4.21: Three unseen identities are fitted with global model on left column, intermediary models on central column, and local models on right column. The global model reaches an interesting fitting accuracy on frontal and neutral faces when it is specialized on this kind of faces. It can actually suffice for many applications. But accuracy is often required for many systems built to provide reliable judgements. The fitting accuracy reached by intermediary and local models is higher, and often close to manual labels thoroughly placed on the picture several times. The vertices of the model are also very smoothly distributed over the facial features what increases the reliability of interesting distance measurements for medical devices for instance.

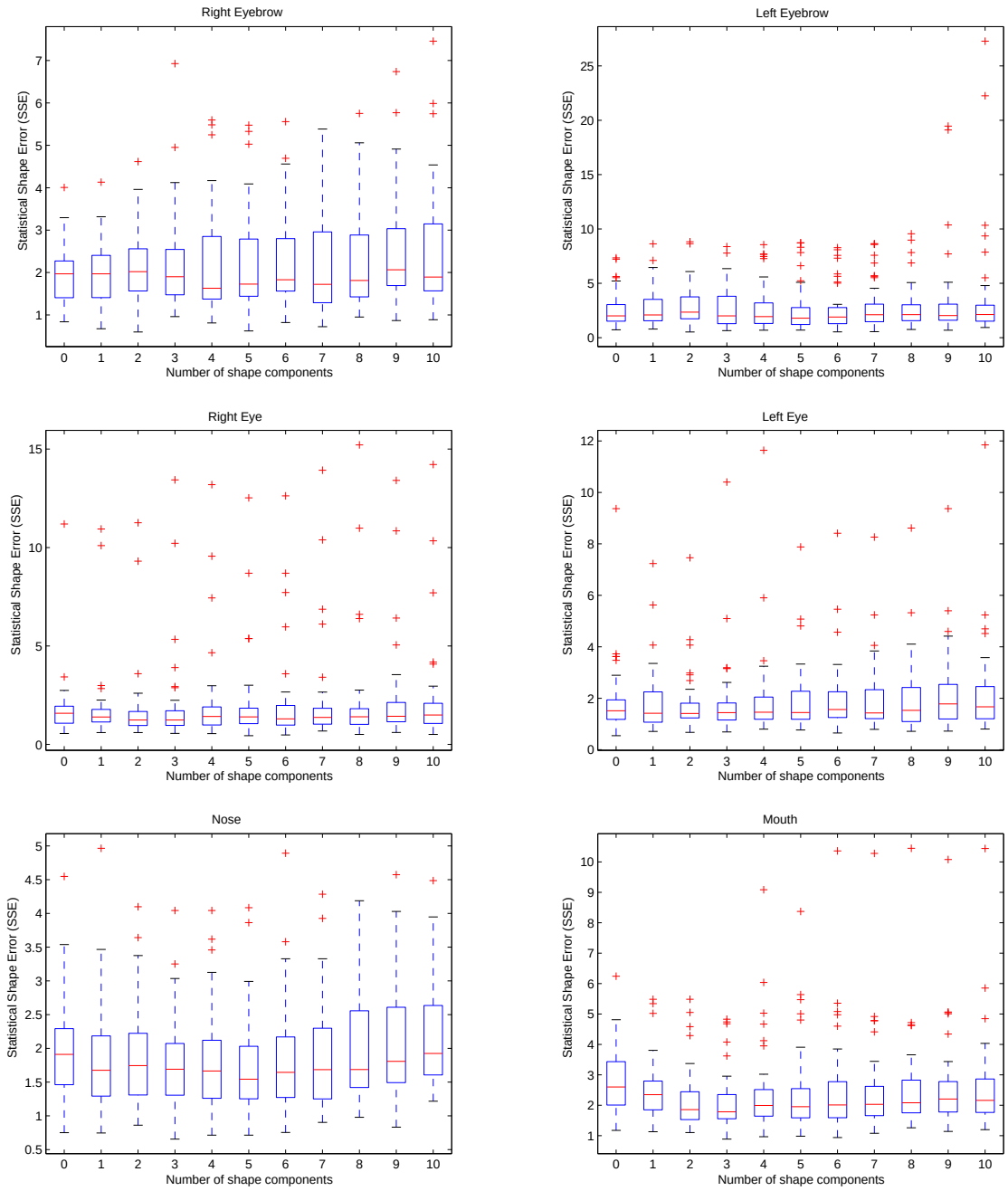


Figure 4.22: Local models fitting results for each feature and for a varying amount of shape components.

clearer the relevant results.

The fact that results are not symetrical from left eyebrow to right eyebrow is due to the inequivalency of left and right feature training set. Indeed, the image labels are usually not symetrical due to the imperfection in face neutral expression and the not always perfect frontal head pose of the training people as well as the possible slight asymetry of the studied faces. One side feature is not obligatorily equivalent to the other in terms of shape variation. Table 4.1 shows the evolution of eyebrow shape

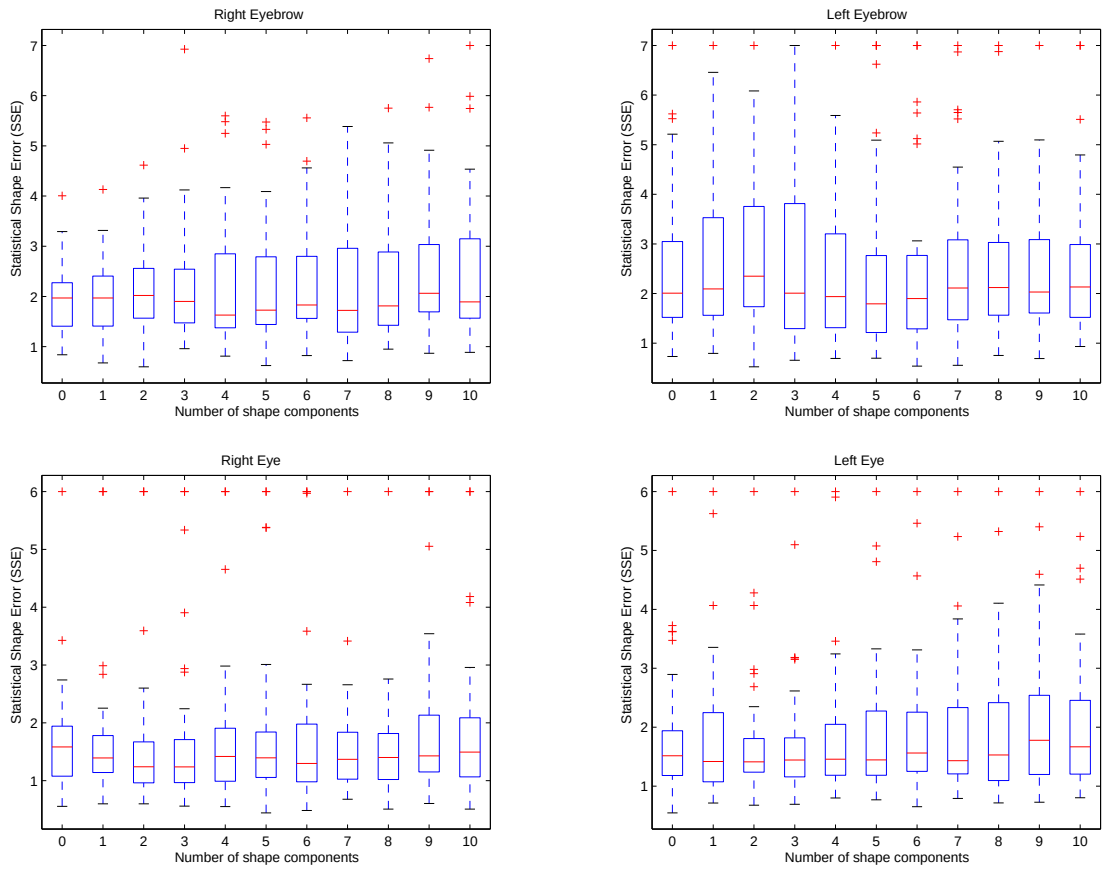


Figure 4.23: Local models fitting results for eyebrow and eye features, for a varying amount of shape components.

variance according to the amount of shape component retained.

Number of Components	0	1	2	3	4	5	6	7	8	9	10
Right Eyebrow	0	66.7	80	87.4	92.3	94.6	96.2	97.3	98.1	98.7	99.2
Left Eyebrow	0	55.3	72.5	81.2	87.9	91.8	95.2	96.6	97.4	98.2	98.9

Table 4.1: Percentage of shape variance retained in right eyebrow local model (second line) and in left eyebrow local model (third line) for a given number of shape components retained (first line).

We can notice that for the same amount of shape variance, the fitting results present the same kind of behaviour. From table 4.1 remark that for $n = 2$ and above, n shape components retained for the right eyebrow model contains about as much variance as $n + 1$ shape components retained for the left eyebrow model. It makes sense to compare the median fitting result for equivalent quantity of variance retained. Figure 4.24 presents this comparison, and the observation of the graph makes us decide to retain a percentage of shape variance close to 92%.

We repeat this study for eye local models. Table 4.4.3.2 presents the correspondence between number of shape component retained and percentage of variance for left and right eye models. The

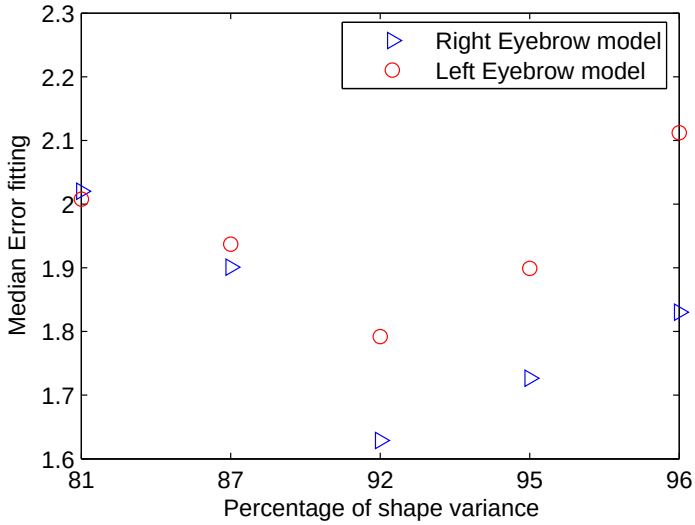


Figure 4.24: Median fitting error for the right and the left eyebrow local models for a range of shape variance between 81 and 96.

difference is small and not particularly meaningful: the choice between two or three components is still not obvious. The optimal shape variance appears to stand between 73% and 85% of variance. To push a bit further toward an answer, we gather all fitting scores obtained with both left and right local models, and plot the new box of whiskers on figure 4.25. Between two and three components, more accuracy seems to be reached with two components, equivalent to about 75% shape variance. We will keep this amount of shape variance for eye models since it also ensures more robustness than three or more components.

Number of Components	0	1	2	3	4	5	6	7	8	9	10
Right Eye	0	51.3	73.4	82.9	89.9	94.1	96.4	98	99.3	99.9	100
Left Eye	0	51	77.1	85.1	90.1	93.9	96.2	97.8	99	99.9	100

Table 4.2: Percentage of shape variance retained in right eye local model (second line) and in left eye local model (third line) for a given number of shape components retained (first line).

We will now stop our decision on the shape variance amounts gathered in the following table.

Eyebrow models	92%
Eye models	75%
Nose model	94%
Mouth model	70%

Table 4.3: Final shape variance percentages retained to be the optimal to build the local models.

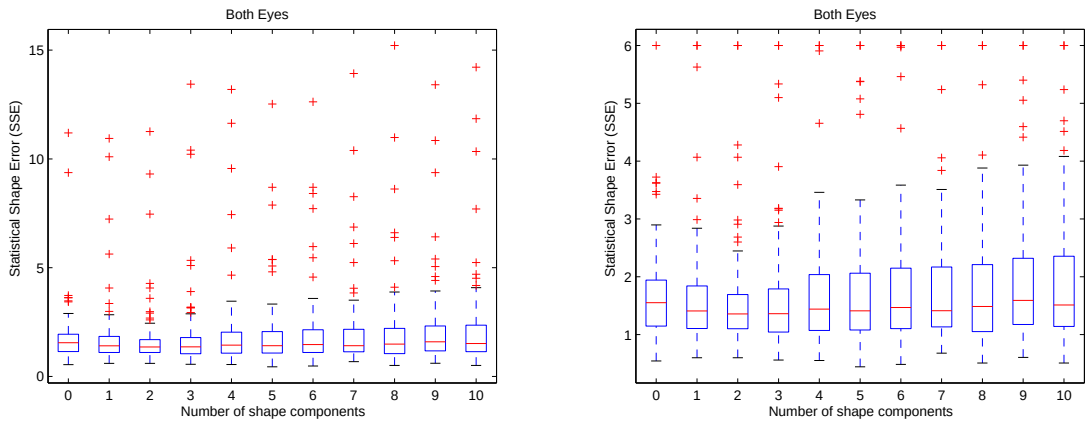


Figure 4.25: Fitting results for both eye local models for a varying amount of shape components.

4.4.3.3 Comparing local model accuracy results to global models and to manual labelling

We determined the amount of shape variance that optimizes the fitting accuracy for each model, intermediary and local ones. The results are obtained from quite a reasonable number of faces but we believe they are meaningful and the statistics we build is consistant enough.

We now want to evaluate the fitting accuracy of our method, against the fitting accuracy we can obtain with a global model when fitting on unseen data. Since our fitting accuracy evaluation method allows to estimate the quality of manual labellings of the human expert, we will also compare those results from global and local models to those of manual labellings.

On Figure 4.26, we compare the results obtained with global model, intermediary model, local model and both worst manual labelling scores and average manual labelling scores.

Figure 4.28 shows the general accuracy comparison considering all points that are common between each step of the coarse-to-fine strategy. We added a box concerning the accuracy results obtained with global refitted data. The global refitted model is obtained by training the global model onto refitted data: the used face data is learnt by an AAM retaining 99% variance of shape and appearance, and is fitted again on the same faces in order to increase the correspondence between vertices among the training data. Introduced in [57], this operation seems to improve the fitting results on unseen faces with respect to the results obtained when training the AAM on once-labelled data. Since we use multiply labeled data for training (the mean of 10 labels to define each vertex), their semantical position on the face is high and should naturally improve the correspondence among data.

We believe that the higher semantical meaning obtained with label statistics is mainly responsible for this improvement. Indeed, these labels have higher semantical meaning since human labellers attempted several times to accurately set them into a given position on each face. The refitting process will displace once-labelled vertices to maximize their cohesion, but it is improbable that these new positions are semantically the most correct ones (although they might usually be improved). Multiply labelled data then constitute a maximum bound to accuracy, which explains the improved results obtained when we train an AAM with these data with respect to those obtained with refitted data.

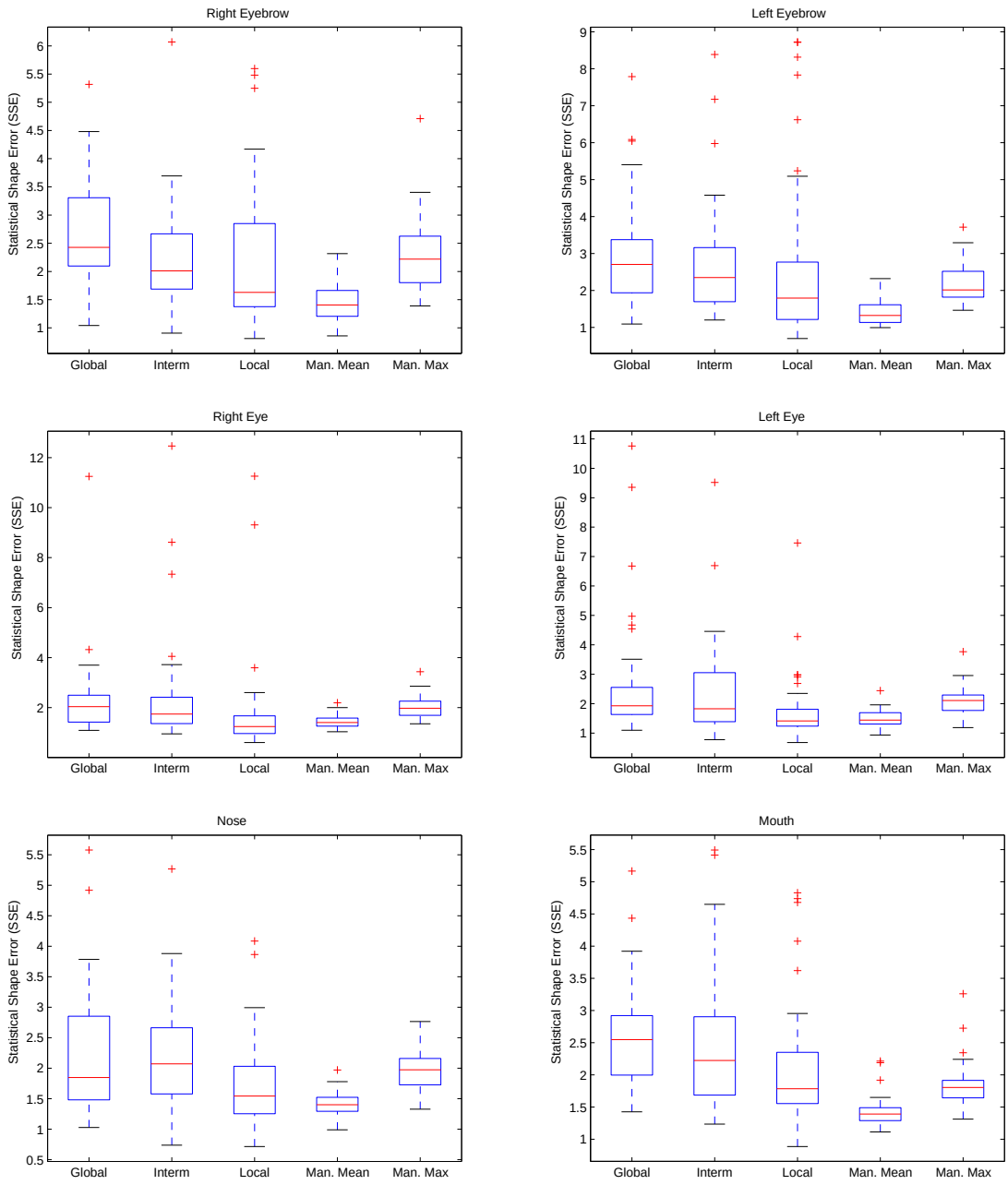


Figure 4.26: Comparison between fitting scores obtained with global model for its best parameterization in shape and appearance variance, (for intermediary models), for local models, and manual labellings, both worst and average scores. Local models often reach a fitting score comparable to manual labelling.

4.4.3.4 On the influence of the training set size

Compared to the several thousands of training images predicted by Gross *et al.* to generalize on unseen faces, the 40 images used for training in this chapter seem very few. We want to understand how could improve the accuracy results with an increase of the training set dimension.

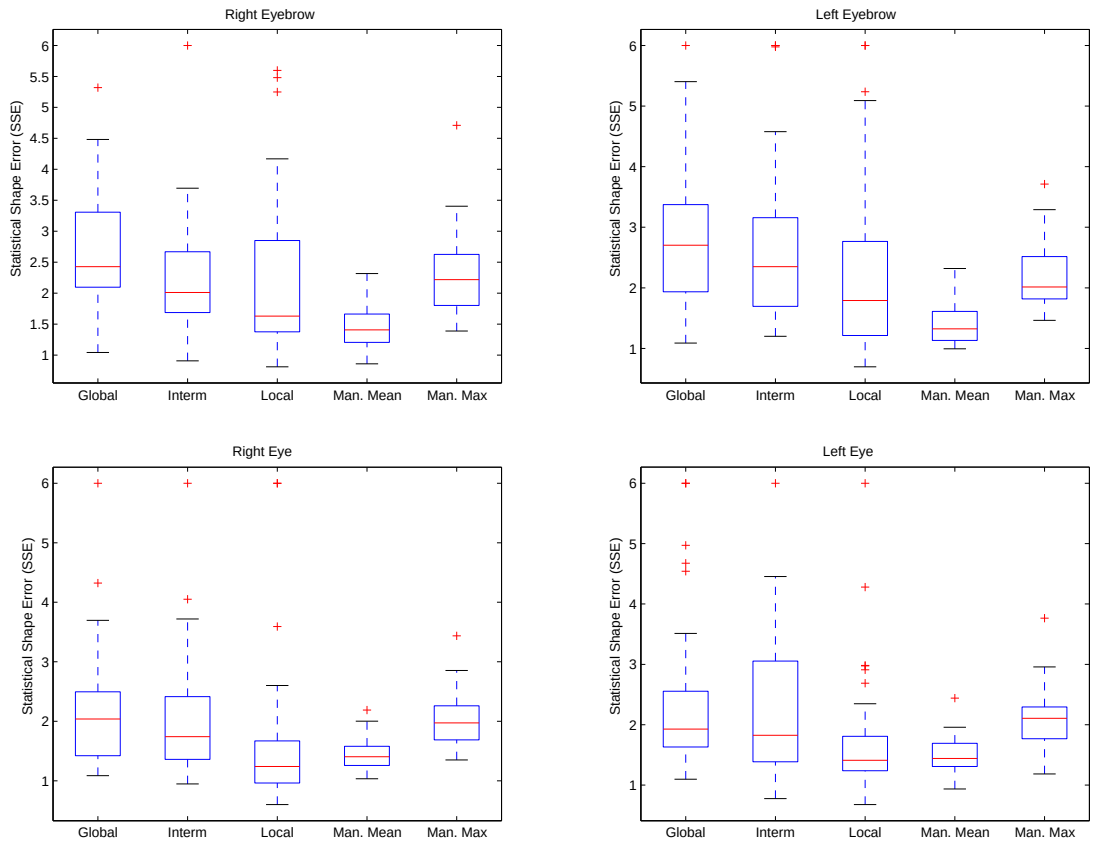


Figure 4.27: Comparison between fitting scores obtained with global model, for intermediary models, for local models, and manual labellings, both worst and average scores. Local models usually reach a fitting score comparable to manual labelling.

On the basis of the 40 images we have, we conduct the following test. For each 40 images, in turn considered for a fitting test with the global to local fitting process, we randomly select N images to train the AAM. We test for N all the values $[1, 3, 5, 10, 20, 39]$. For each N tested, we thus make 40 fitting tests for which we score the results with the SSE. On figure 4.29 we represent the median SSE score for each N tested. Somehow surprisingly, 39 images used for training do not seem to be a small amount. It actually seems that more training pictures would not result in high accuracy improvement. The test would be worth doing, but the extra required data should be thoroughly labelled 10 times too.

4.5 Conclusion

In this chapter we addressed the problem of fitting an AAM on unseen faces. An AAM can be extremely good at fitting onto data that it learnt, but we observed that fitting onto unseen faces was prone to diverge or at least to lack of accuracy and robustness. The best performances that can be obtained in the unseen context are obtained when we restrict the learning face data to one pose and one expression, and when many identities (morphologies) are learnt. We investigated the fitting performance in this context and clearly reveal the reason for its limitation. The appearance statistics

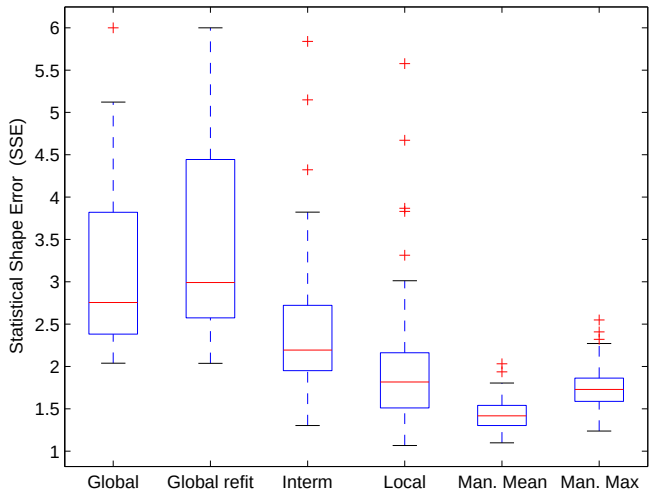


Figure 4.28: Comparison between fitting scores obtained with global model, refitted global model, intermediary models, for local models, and manual labellings, both worst and average scores. Local models usually reach a fitting score comparable to manual labelling.

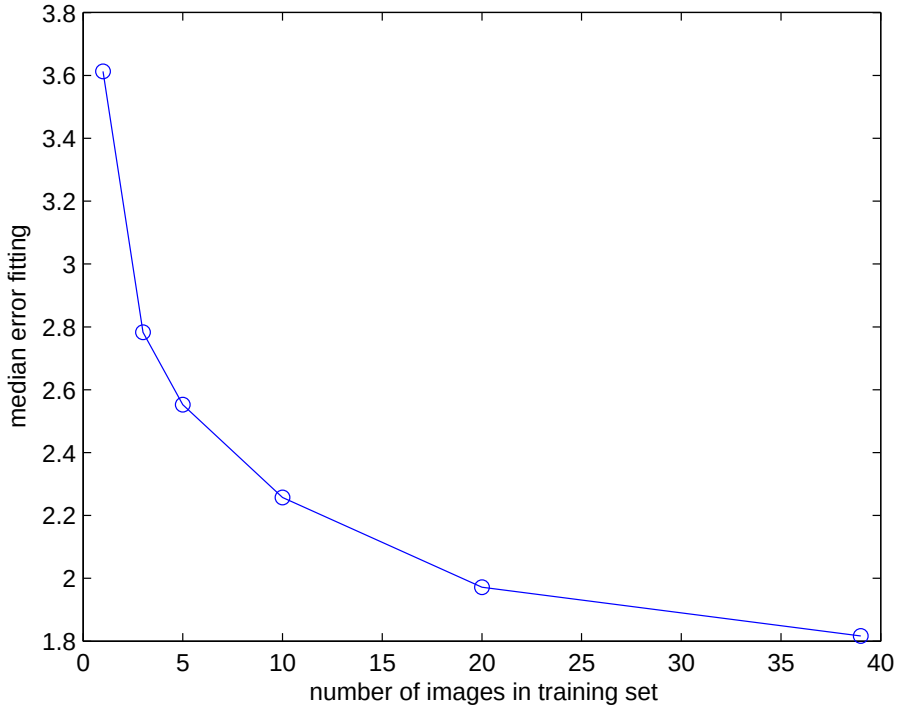


Figure 4.29: For different training set size, the median fitting accuracy among 40 tests is represented. 40 image training is a number that optimizes the trade-off between speed (more training images would increase the number of components retained in the AAM) and accuracy (the curve shows that it would not be much improved for a larger training set).

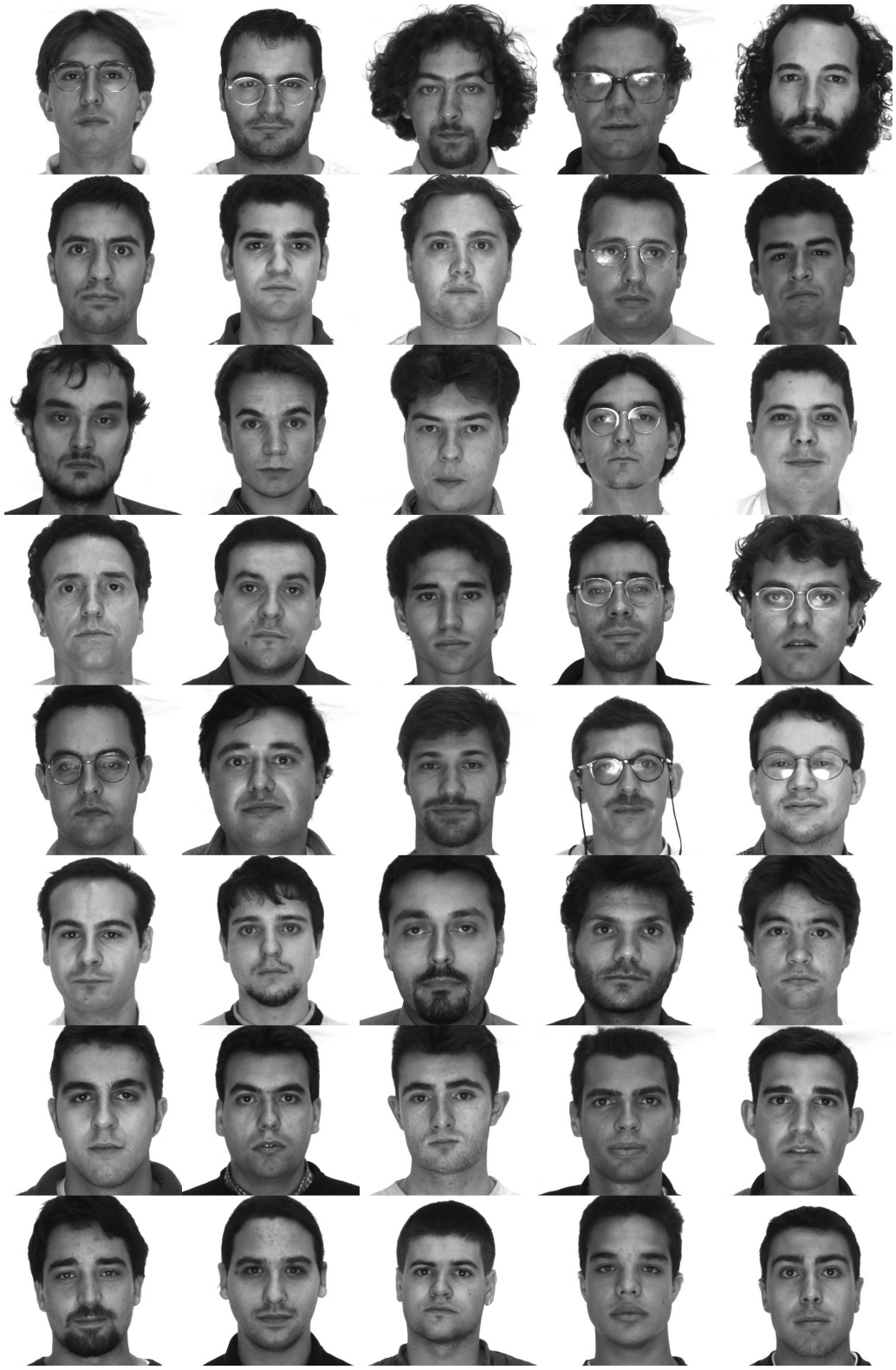


Figure 4.30: The 40 AR database frontal and neutral faces used for tests.

of the AAM cannot generally fully explain the unseen data. However, the optimization process keeps minimizing the remaining error due to this discrepancy between the input image and the appearance statistics, what results in a displacement and distortion of the model, even larger when the model is very deformable (when high shape variance is kept to build the model). For 39 image training, the best overall fitting accuracy results on unseen faces were obtained by keeping 60% of shape variance in a way the model can deform and adapt to the unseen faces without bending too much under the pressure of unexplained visual input data, and the maximum appearance variance (any amount above 90% is fine). The proposed tests were all conducted through the *Statistical Shape Error* we propose to objectively judge upon the fitting accuracy.

To increase the expressivity of the appearance statistics, we explained conceptually and validate experimentally the fact that an AAM covering a smaller area of the face can better reconstruct the unseen face appearance that a global one (over the surface they have in common).

To increase the fitting accuracy we therefore make use of a collection of local models that effectively allow to obtain higher fitting accuracy on unseen faces. Unfortunately, due to there small size, these models should be very well initialized on the input picture. To ensure this, we rely on a 3 step coarse-to-fine strategy. Starting from a global model, quite robust to bad initialization but providing a rather modest fitting accuracy, we initialize a set of intermediary models that can in turn improve the fitting accuracy. As a last step, the intermediary models give local AAMs that are designed to fit onto one facial feature each. The fitting accuracy reached by the local models is generally comparable to manual labelling accuracy as can be asserted with the Statistical Shape Error.

Besides, we show that the statistical annotations (the mean of the ten manual labels place on a face image to define each vertex of the face) used for training the AAM also improve the fitting accuracy. Indeed, the results obtained with the global model trained this way outperforms the accuracy reached by the global model trained with refitted data. A global model trained on refitted data itself actually show in [57, 113] to outperform the fitting accuracy reached by a model trained on data manual labelled once only.

As a perspective, we guess that a solution to build a global AAM combining a partition of independent appearance statistics can be build. Instead of using a set of different models, the same properties could be gathered in only one global AAM.

Most of the important points of this chapter were presented in [87].

In the next chapter we will develop more the concept of specialized and local models to deal with pose and expression variation on unseen faces.

Chapter 5

Expression and Pose Retrieval on Unseen Faces

5.1 Introduction

Chapter 3 taught us that we cannot count on one generic AAM to fit an unseen face displaying various expressions at different head poses. However, fitting results on unseen faces can be pretty accurate when the AAM gets specialized on frontal and neutral expression as we have seen it in chapter 4. Moreover, we proposed a way to achieve even better fitting accuracy through the use of segmented models. It seems that a solution (is there any other?) to use AAMs reliably for fitting unseen faces is to specialize the model as much as it can be. In the previous chapter, the studied pose was frontal and expression neutral, but we can logically expect to observe similar results for any other given couple of pose and expression.

In this chapter we develop this philosophy of specialized AAMs. We want to see whether it can be pushed further and used to fit unseen faces under varying poses and facial deformations. The problem can be reconsidered in the following way: instead of considering that a face can vary in pose, expression and identity, what makes it a very complex object, we will here consider that for each pose and each expression, the face is one specific object that only has a variability in morphology. Under such a consideration, if we want to deal with any pose and expressions, we will have to model each of their possible combinations separately since they represent different objects.

We propose to build a collection of AAMs, where each one only learns one pose and one expression, for several identities. We will train various AAMs, one for each couple pose/expression we wish to retrieve, and we build a pool of pose/expression specific AAMs.

On an input frame presenting an unseen face, we will need to know which AAM to choose from the pool. The problem is that we do not know what is the current pose and expression of a face. What we propose is to fit all AAMs onto the processed frame, and to select as the winner the one that presents the smallest average residual error. Such a winner AAM should stick to the face better than all the others, and it should also be the most appropriated to fit the current facial display. We should then be able to gather two important abilities in our system, to classify the current deformation, and also to retrieve the shape of each facial feature as we expect the winner AAM to fit the features with accuracy. This would present an advantage with respect to machine learning based classifiers only able of the first ability.

In the following, we propose to verify whether the winner AAM from a pool of tested AAMs correctly classifies the deformation, and fits accurately on the facial features.

In 5.2 we present the database we built to allow the tests. In 5.3 and 5.4 the protocol and tests are

n° deformation	description	part of the face
0	neutral	upper
1	neutral	lower
2	low smile	lower
3	mid smile	lower
4	high smile	lower
5	open smile	lower
6	frown eyebrows	upper
7	raised eyebrows, eyes wide open	upper
8	mid closed eyes	upper
9	closed eyes	upper
10	watch left	upper
11	watch right	upper
12	watch up left	upper
13	watch up right	upper
14	watch down left	upper
15	watch down right	upper
16	watch up	upper
17	watch down	upper
18	low open mouth	lower
19	mid open mouth	lower
20	sad mouth	lower

Table 5.1: Table of the facial deformations that compose our pose and expression database.

presented as well as the evaluations.

5.2 The Pose and Expression Database

None of the publicly available face database can help to test the AAM performance in case of pose and expression variations. We often find databases where the face illumination is not completely controlled and repeated identically on the whole database. Sometimes the picture quality is pretty poor. Or the problem can also be that not enough poses and/or facial deformations are represented, or their intensity is not properly controlled.

We decided to build our own database for the purpose.

5.2.1 Description of the database

The database is composed of various identities (we have 9 identities for the moment), acquired under 3 different poses (0°, 10°, 20°) and under homogeneous illumination. For each pose, each person performed a sequence of 21 facial deformations, each concerning the upper or lower part of the face (or both for the neutral expression). Table 5.1 lists these deformations and their associated number.

Each picture of the database presents one useful upper or lower facial deformation (except for the neutral expression which is represented by a same picture for the lower and upper part). We will then have to model upper and lower deformations separately. This offers two main advantages: (i) we can separate the analysis of lower and upper facial parts, what results in less combinations to test, (ii) the

AAMs will have to model only the upper or the lower part of the face, what will result in potentially higher fitting accuracy on unseen faces since such models are local and not global. However we will not have the possibility to build a collection of global models presenting all the possible combinations of lower and upper deformations. Such models would be useful to initialize the local models robustly. Anyway, a different manner should be found to initialize the local upper and lower models on a face since the use of global models representing all possible combinations of upper and lower deformations would spoil the advantage stated in (i).

For this study, we will not consider the problem of initialization of the models on the face (what should actually easily find a solution, especially in the case of facial video analysis where an intialization should only be done once, on the first frame). We first want to understand if we can retrieve the pose and the expression of an unseen face with specialized AAMs. If results are positive, then it will make sense to investigate a solution to initialize these models properly. We will come back on the test protocol in 5.3.

For each category, we also wanted some deformations to be very different, and some to be very close one to another. The test should then lead to a better understanding of the AAM ability to best fit the deformation it is trained for, despite the closeness between some tested deformations.

5.2.2 Difficulties encountered

The database construction requires a lot of time. Each identity included in the database represents about 3 hours of work. First the person must train to perform the sequence of deformations. Since we only had one camera (Canon E80 reflex) it is necessary to take a sequence of pictures from each angle. Taking one first time all the pictures from the three different angles is usually a good way to train the person to perform the expressions correctly. A second shot of the deformations from every angulations is usually better in terms of quality of the deformations and control of their intensity. Once all pictures are taken, they must be labelled manually. Here (for obvious time reasons) we did not label each picture several times to improve the ground truth shape, but we tried to label them once with accuracy to obtain improved results.

Despite all the care brought during the database construction, we could observe that it was:

- hard to make people keeping the correct pose always, especially for some deformations like watching on the sides where it happens spontaneously that people cannot help but turn the head a little,
- hard to control the intensity of the performed deformation,
- sometimes difficult for the people to perform certain deformations: they sometimes needed some training time to rehearse the muscles linked to a special deformation
- sometimes difficult to define if the way people perform a deformation is right or wrong due to some confusing morphologies.

The accuracy of the data contained in the database is of major importance for the classification correctness. We will discuss this question in 5.3

The problem of head pose could be solved by blocking the head rigidly. But this would make the shooting session even more involving for the participants, and they would probably feel even less free to display the expressions and a major training time would be necessary.

The problem control the deformation intensity is not a trivial one. It is hard to visually understand if the right intensity is performed. It is also basically hard to determine a certain intensity, and after to determine when a person is precisely displaying the expression at this right intensity. What makes the judgement hard is also the variety of morphologies and personal way to perform a deformation.

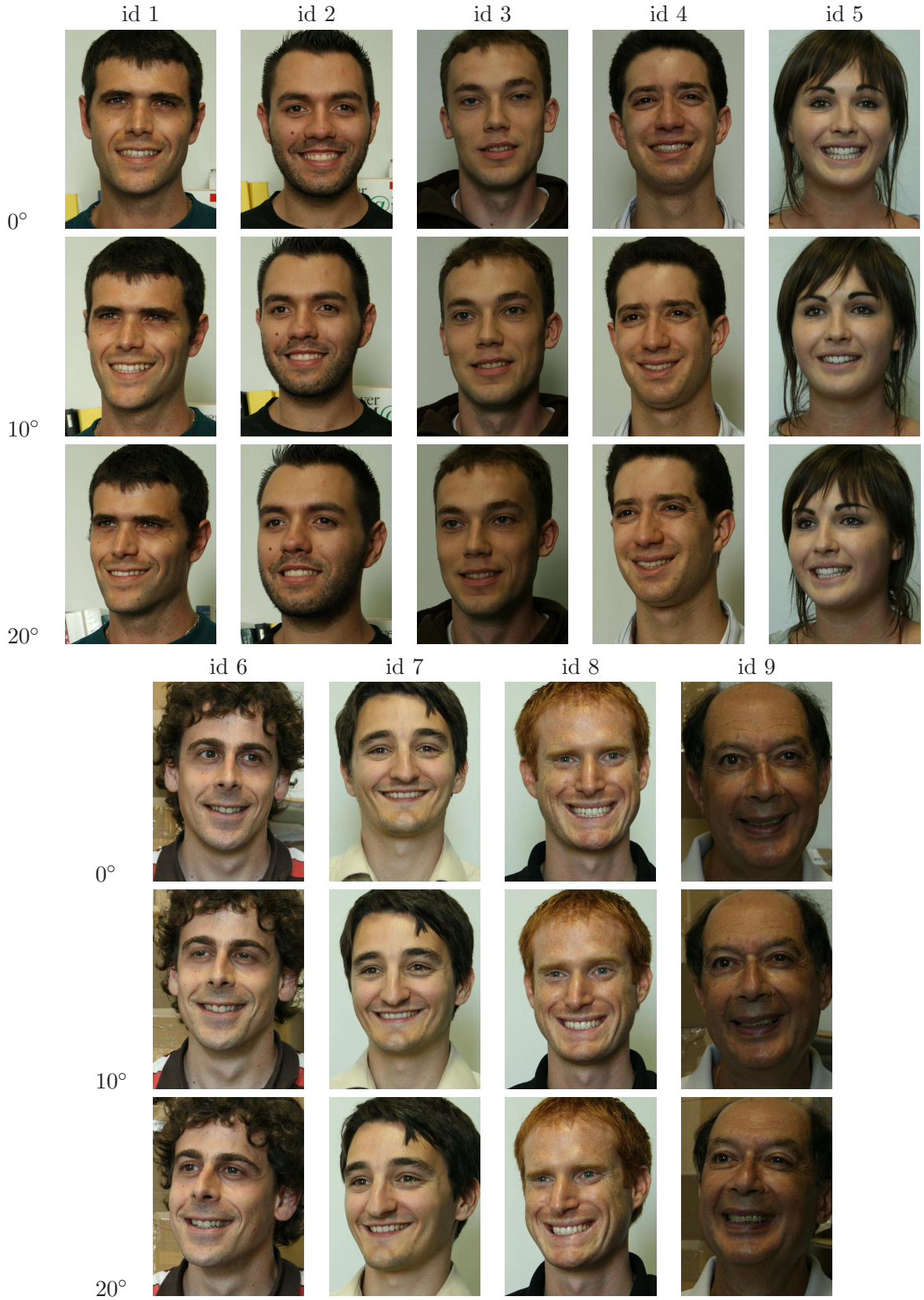


Figure 5.1: All 9 identities of the database are presented here for all poses and displaying the smile deformation (expression n°9).

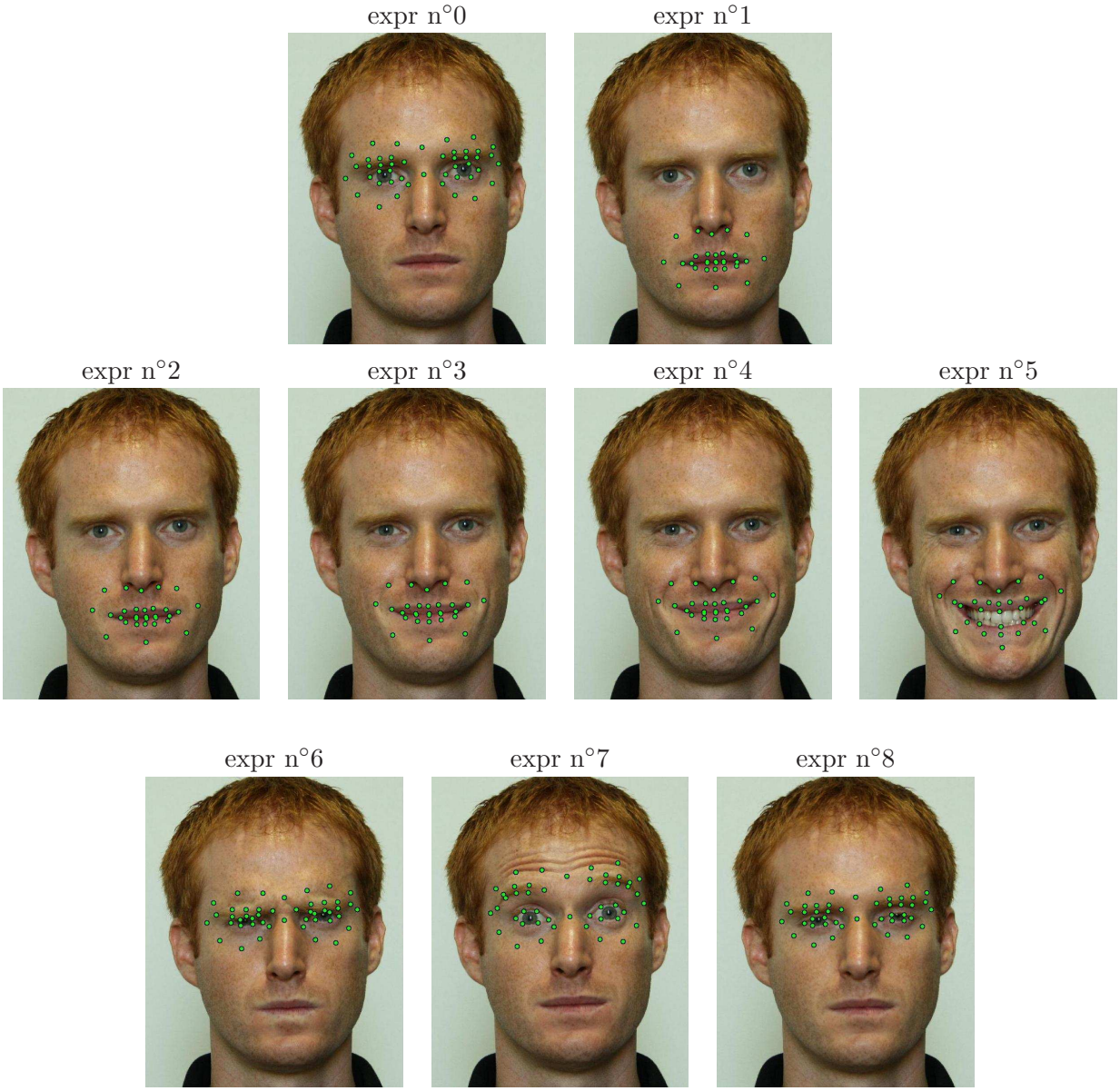


Figure 5.2: Expressions (from 0 to 8) represented in the database with manual landmarks representing the vertice used for training or used to initialize the fitting process when this identity is used for test.

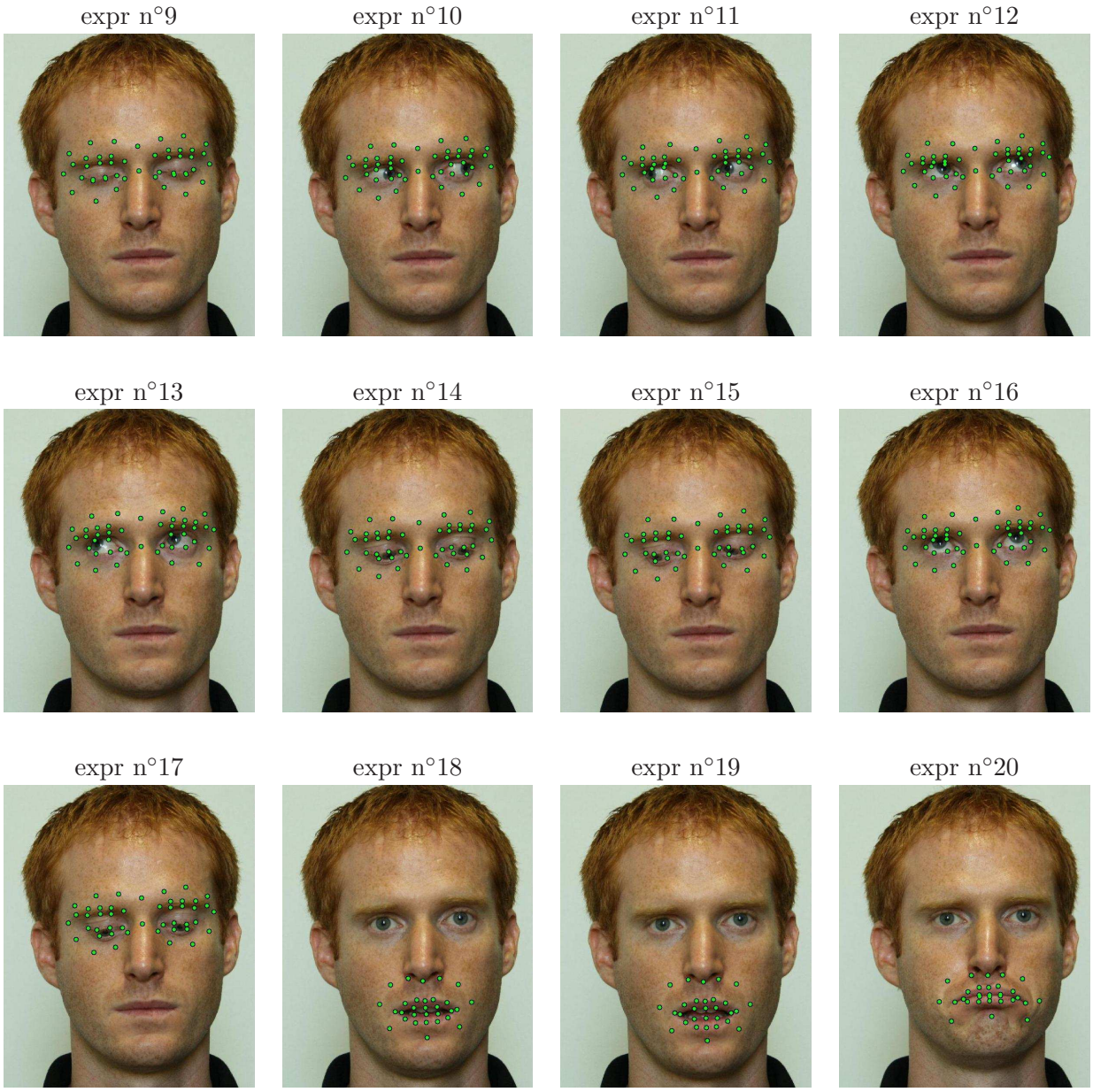


Figure 5.3: Expressions (from 9 to 20) represented in the database with manual landmarks representing the vertice used for training or used to initialize the fitting process when this identity is used for test.

5.3 Protocol

Since our database gathers a modest amount of identities (9 people), we will conduct a preliminary test. A more complete one should be done with more identities in the database. However, it is probably not a bad idea to draw some conclusion without extensive data in the database since, as we will see, the results are strongly linked to the quality of the database that should be thought and built again with extra thorough.

The test we propose here consists to train one AAM for each possible combination of pose/deformation available. We thus build 63 AAMs for both upper and lower deformations.

In this test we will consider various and growing amounts of training data to check the influence of the training set size on the results. The first test is conducted in a leave-one-identity-out way on the first 4 identities of the database. Each specialized AAM is trained over three identities, and is meant to be fitted onto the remaining non-learnt person pictures. 63 pictures (21 expressions for 3 different poses) must be tested for each unseen identity. As said previously, they are very well initialized: we give the manual labels of the test picture as initial shape for the AAMs. On each of the 8 pictures concerning a lower deformation we run all 24 concerned AAMs (8 deformations for 3 poses). On each of the 13 pictures concerning an upper deformation we run all 39 concerned AAMs (13 deformations for 3 poses). The fitting process lasts 20 iterations for each AAM. After the last iteration, we consider the error image and we compute the average absolute residual error commited by the process.

For one test picture presenting a particular deformation, we will select the winner, or best, AAM as the one that obtained the smallest average absolute residual error. We obviously hope that this winner/best AAM's pose and deformation it represents effectively matches the deformation and pose displayed on the test picture. We also hope that this AAM fits accurately the analyzed features.

As a first analysis of the performance we can obtain through such a protocol, we will observe whether the AAM is correct in terms of correspondence with the current pose and deformation displayed on the picture, and good in terms of fitting accuracy.

Other two tests are conducted with two higher amounts of training data: 5 image training (the first 6 identities of the database are considered), and 8 image training (all 9 identities are considered) are used consecutively. Always in a leave-one-identity-out way, we perform the fitting tests on the same first 4 identities of the database.

We have a strong idea behind this test. The presented process can be very slow since each picture of an expressive sequence should be tested with all available AAMs. Moreover, we could broaden the set of poses and expressions tested, further slowing down the process. But if this test shows a positive exit, that means that a new person could be analysed offline, on sequences aquired and stored in memory. Our long, but automatical process could therefore accurately label its face images, also as label them in terms of facial deformations to associate to a configuration of labelling points. A fast person-specific AAM (or a set of person-specific AAMs) could thus automatically be built. After this long learning stage, the tracking could be fast on this person, and the classification of the deformation would be straight-forward since all of his facial configurations would be pre-labelled in terms of pose and deformation and these data would already be decorrelated.

5.4 Evaluation

In a practical case, the protocol should be completed in order to deal with both side rotations. Either the training images concerning non-frontal poses should be inverted to build an extra set of $-10°$ and $-20°$ AAMs, or the input test picture should be inverted and then tested for both orientations.

We will here simply observe the results obtained on the frontal case, tested with the (non-inverted) AAMs presented at the previous section. We briefly had a look at the results obtained on test pictures with pose at 10° and 20° and concluded they were similar to the results obtained for the frontal case that we present here. The results we present here are preliminary and they do not mean to be fully complete. But we want to draw a first consistent conclusion on this test, to know if the investigation is worth being followed in perspective works. This will be discussed in conclusion of this chapter.

As said in the previous part, we want to see whether the winner AAM is correct (matching its pose and deformation to the one displayed on the test picture) and presents a good fitting accuracy.

During the tests we observed four cases:

1. the best/winner AAM is correct and the fitting is good in accuracy,
2. the best AAM is incorrect but the fitting is good in accuracy,
3. the best AAM is correct but the fitting is bad in accuracy,
4. the best AAM is incorrect and the fitting is bad in accuracy.

We give the following interpretation to each of these four observed cases:

1. the feature is well known by the correct AAM that results best performing on it,
2. the feature is usually well known by the correct AAM but a best fitting score is obtained by a close AAM, usually because its range of deformation well encompasses the fitted one (problem of overlapping deformation intensities in the database), and also due to some particularity in the person morphology or way to express the deformation (that can also be due to inaccuracies in the database construction), bringing it closer to a deformation expressed by another AAM.
3. this usually happens when the face feature is badly known by the correct AAM and by the others too, what results in a *happy last resort* since the correct AAM wins, but still the fitting is not accurate. This particular face should be included in the training set since it brings some unknown information.
4. the feature is not well known, maybe particular in morphology or/and way to express the deformation. The AAM is not chosen correctly nor the winner is accurate. Also in this case, the shape and/or appearance of this face feature is missing in the training set and should be added to it.

The correctness of the best AAM is an objective criteria since it is claimed when both pose and deformation represented by this AAM match the pose and deformation displayed on the picture.

We judged visually on the fitting accuracy of the best AAM. This criteria is thus subjective. To allow the reader to see how demanding we were for considering the fitting to be accurate, some visual results of fitting considered to be good or bad in fitting accuracy will be showed in the following.

We counted the occurences of the four observed cases for 4 tested identities, and for 3 different amount of training pictures used to build the pool of AAMs.

They are reported in Table 5.3 in a way that is explained by the example Table 5.2.

The results are very convincing for identities 1 and 4. The test is also coherent for identity 2. Results on identity 3 can appear a bit curious, but they should follow the results of the other 3 identities by adding some identities closer to this one in the training set.

For identity 1, something looks evident: the number of accurate fittings when the training set contains 8 images. The improvement is clear between (3 or) 5 and 8 training images. This is not the number of extra images added that counts really, but rather the morphology of the new identities newly inserted in the training set. One new person representing the missing shape or the missing appearance of the unseen test face can dramatically improve the fitting accuracy results on this face.

fit/aam	good	bad	
good	n_{case_1}	n_{case_2}	$n_{case_1} + n_{case_2}$
bad	n_{case_3}	n_{case_4}	$n_{case_3} + n_{case_4}$
	$n_{case_1} + n_{case_3}$	$n_{case_2} + n_{case_4}$	

Table 5.2: Illustration of how results are presented in Table 5.3. n_{case_i} represents the number of occurances of the case i.

	3 training images				5 training images				8 training images			
Id1	fit/aam	good	bad		fit/aam	good	bad		fit/aam	good	bad	
	good	4	7	11	good	7	3	10	good	11	9	20
	bad	1	9	10	bad	2	9	11	bad	0	1	1
		5	16			9	12			11	10	
Id2	fit/aam	good	bad		fit/aam	good	bad		fit/aam	good	bad	
	good	8	5	13	good	13	1	14	good	13	1	14
	bad	2	6	8	bad	2	5	7	bad	1	6	7
		10	11			15	6			14	7	
Id3	fit/aam	good	bad		fit/aam	good	bad		fit/aam	good	bad	
	good	13	3	16	good	9	5	14	good	12	3	15
	bad	3	2	5	bad	2	5	7	bad	0	6	6
		16	5			11	10			12	9	
Id4	fit/aam	good	bad		fit/aam	good	bad		fit/aam	good	bad	
	good	7	4	11	good	7	3	10	good	14	4	18
	bad	7	3	10	bad	5	6	11	bad	1	2	3
		14	7			12	9			15	6	

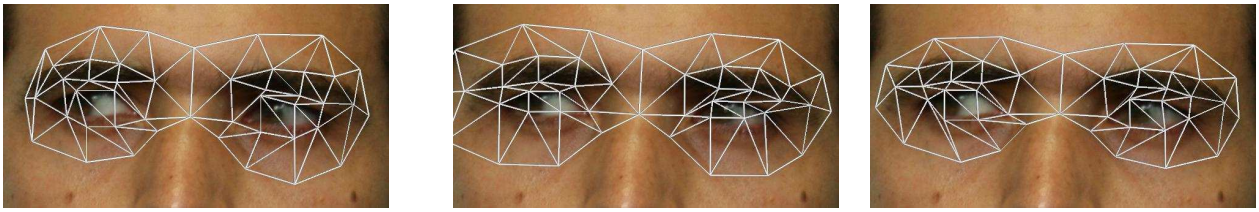
Table 5.3: Number of occurances of each of the four observed cases. They are reported for identities 1 to 4 and for three different dimensions of the training set. These results are presented for the test pictures of pose 0°.

This can be understood thanks to the following explanation. If a person is not well fitted by our process, it means that his face is far from the faces that already belong to the training set. This face contains a lot of new information with respect to the current training set. It therefore should be added to the training set to make it able to generate more faces. Should this particular face be added, the process would fit that face greatly: this remark stands just to exagerate on the fact that the fitted face should somehow be well represented by the training data if we expect to obtain good results on it.

For identity 1 having particularly low eyebrows, all upper feature fitting results were bad until identity 8 were introduced in the training set, showing some identically low eyebrows too. Some illustrative results are shown on Figures 5.4, 5.5, 5.6.

The same observations can be done for identity 4 for the lower face too. Some illustrative results are shown on Figures 5.7, 5.8.

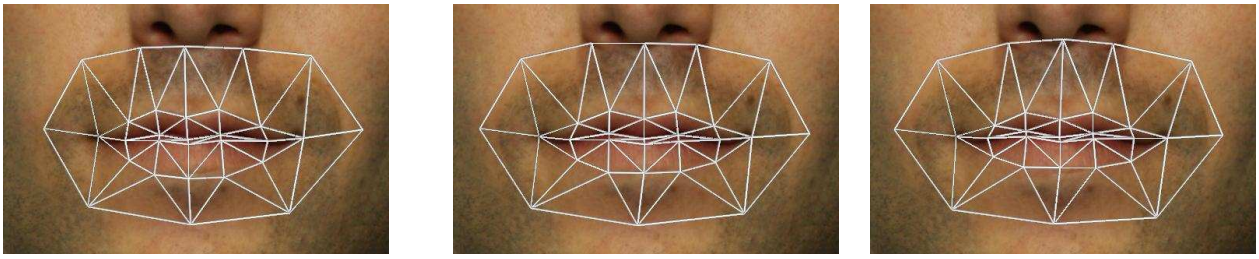
TrainingSetDim 3 | TSDim 5 | TSDim 8



BestAAM: expr.13 pose 0° (wrong) Judged: bad fitting | expr.6 pose 0° (wrong) bad fitting | expr.11 pose 0° (correct) good fitting

Figure 5.4: On id.1, expression 11 is observed at pose 0°.

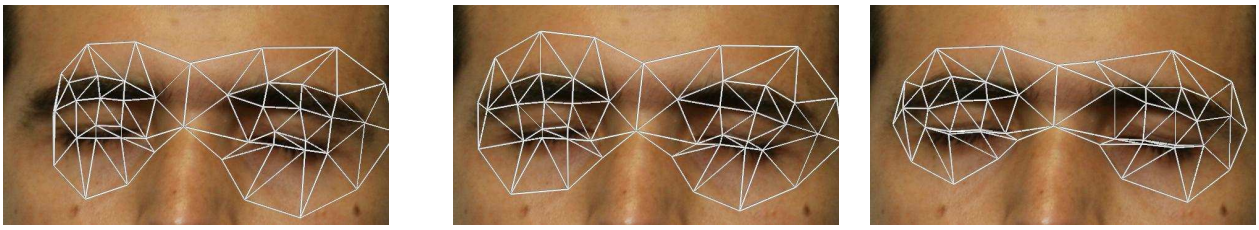
TrainingSetDim 3 | TSDim 5 | TSDim 8



BestAAM: expr.2 pose 10° (wrong) Judged: bad fitting | expr.2 pose 0° (wrong) good fitting | expr.2 pose 0° (wrong) good fitting

Figure 5.5: On id.1, expression 3 is observed at pose 0°.

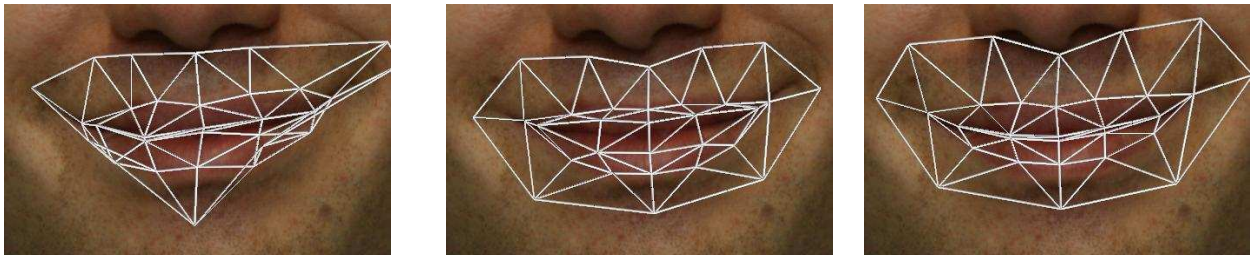
TrainingSetDim 3 | TSDim 5 | TSDim 8



BestAAM: expr.8 pose 20° (wrong) Judged: bad fitting | expr.8 pose 10° (wrong) bad fitting | expr.9 pose 10° (wrong) good fitting

Figure 5.6: On id.1, expression 9 is observed at pose 0°.

TrainingSetDim 3 | TSDim 5 | TSDim 8



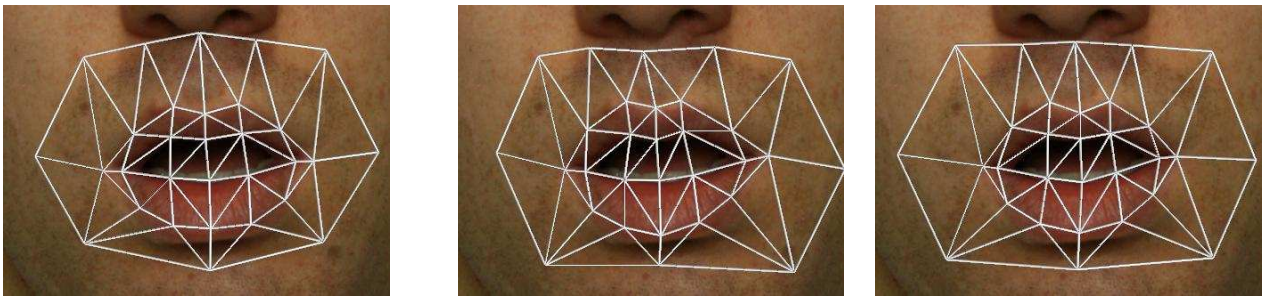
BestAAM: expr.3 pose 0° (wrong)
Judged: bad fitting

expr.5 pose 0° (wrong)
bad fitting

expr.3 pose 0° (wrong)
good fitting

Figure 5.7: On id.4, expression 4 is observed at pose 0°.

TrainingSetDim 3 | TSDim 5 | TSDim 8



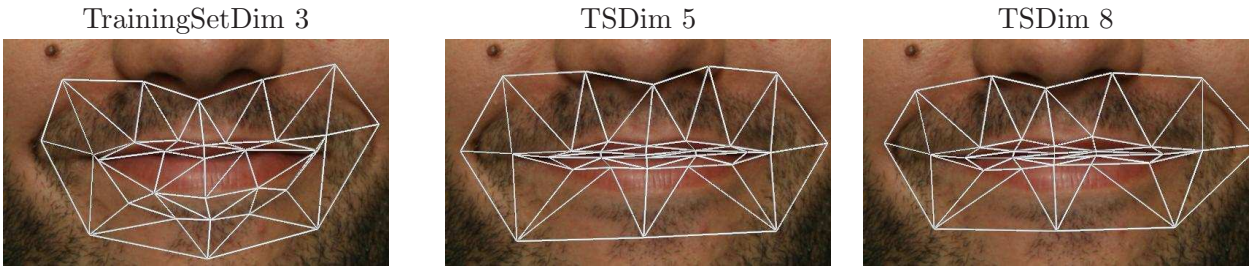
BestAAM: expr.19 pose 0° (correct)
Judged: bad fitting

expr.19 pose 10° (wrong)
bad fitting

expr.3 pose 0° (correct)
good fitting

Figure 5.8: On id.4, expression 19 is observed at pose 0°.

For identity 2 having particular mouth and a particular way to smile, the results keep stable. This is because none of the faces from identity 5 to identity 9 is representative for it. An example of this observation can be seen on Figure 5.9. Among identities 1, 3 and 4, some are actually very representative of the upper feature of identity 2 since results obtained on him for as few as 3 training images are very good in terms of accuracy. An example of this observation can be seen on Figure 5.10.

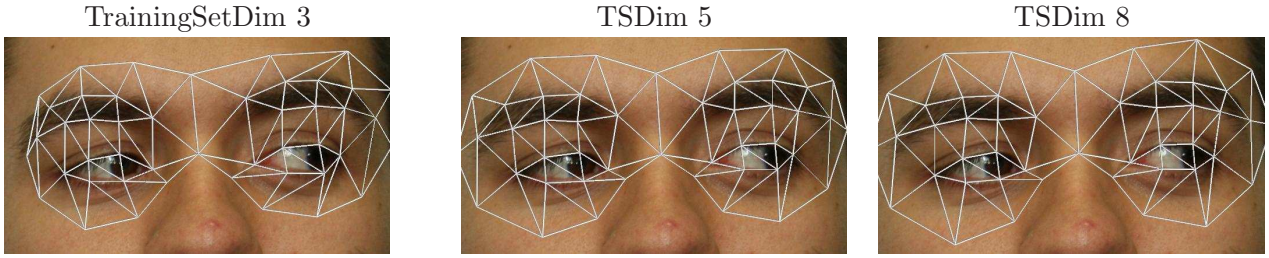


BestAAM: expr.5 pose 0° (wrong)
Judged: bad fitting

expr.4 pose 0° (correct)
bad fitting

expr.4 pose 10° (wrong)
bad fitting

Figure 5.9: On id.2, expression 4 is observed at pose 0°.



BestAAM: expr.10 pose 10° (wrong)
Judged: good fitting

expr.10 pose 10° (correct)
good fitting

expr.10 pose 0° (correct)
good fitting

Figure 5.10: On id.2, expression 10 is observed at pose 0°.

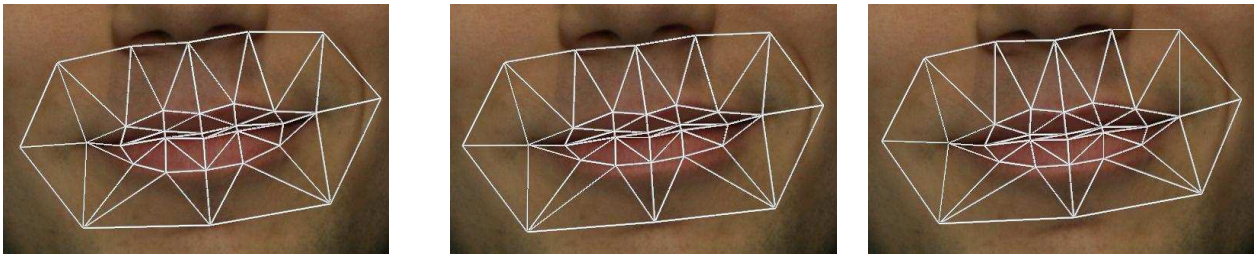
Interpretation is difficult to give for identity 3 who starts with (very) good results for a training set of 3 people, and stays stable (even losing a bit) in fitting accuracy, whereas its number of correct classifications decreases when increases the training set dimension. Some illustrations are given on Figure 5.11.

Some observations helped to explain this partly, and Figure 5.12 is a good example of them.

With only 3 images in the training set, the model is still a bit rigid and have few deformation possibilities to express shapes. It happens here that this rather rigid shape well corresponds to the identity 3's face features, but maybe the appearance does not completely. Then the model is eager to leave its position, but can't since it is too rigid. More deformability effectively leads it to leave this position. Figure 5.12 well illustrates this fact, which is also mainly due to the not strict frontality of the head. The picture shot should be done again with more care.

A higher training set dimension would complete this answer, and we could see if the behaviour of identity 3 turns to be similar to other tested identities if some more representing appearance were included in the training set. Maybe the problem simply comes from the inaccuracy during the picture session with this person.

TrainingSetDim 3 | TSDim 5 | TSDim 8



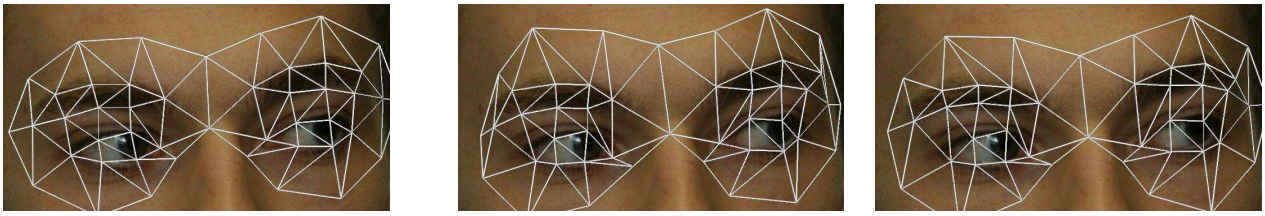
BestAAM: expr.3 pose 0° (correct)
Judged: good fitting

expr.3 pose 0° (correct)
bad fitting

expr.2 pose 0° (wrong)
bad fitting

Figure 5.11: On id.3, expression 3 is observed at pose 0°.

TrainingSetDim 3 | TSDim 5 | TSDim 8



BestAAM: expr.10 pose 0° (correct)
Judged: good fitting

expr.12 pose 0° (wrong)
bad fitting

expr.12 pose 0° (wrong)
bad fitting

Figure 5.12: On id.3, expression 10 is observed at pose 0°.

Concerning the classification correctness, their amount mainly increases when increases the training set dimension, but we should not expect them to tend toward 100% of correctness.

When the incorrect AAM is chosen, it is very often an AAM of deformation which is close to the correct one. The closeness of some deformations makes confusion happenning between best AAMs. This is very likely to happen due to the difficulty to build a facial database that strictly no overlap between the represented close but distinct deformations. The confusion sometimes comes from the pose too, but this does not happen very often. Confusion rather comes from close deformations.

With an increasing dimension of the training set, we would expect to observe a decrease of absurd cases of uncoherent AAM chosen as winner. If the database is built with more accuracy, and poses and deformations are well separated (despite being close), we would expect the correct classification results to increase. Otherwise, we will not expect surprisingly high correct classification rate, but rather that the winner AAM is correct *or* coherent with the correct one.

Probably the best way to build the database would be to shoot a large amount of pictures, and then at last, sort them, deciding which one represent the correct deformation at the right intensity. But the problem of determining the right intensity on each face is a difficult one. Some more studies should be led to reach to a settled accordance of its definition in the facial expression analysis community.

5.5 Conclusion

In this chapter we have presented a first step toward a framework to automatically retrieve facial feature deformations as well as the head pose on unseen faces. The method relies on the construction of a full set of independent AAMs designed to respectively represent one pose and one expression. On an image representing an unseen face displaying a particular facial deformation under a particular head pose, we fit all available AAMs in turn. The best AAM fit corresponds to the correct AAM class (the right pose and deformation are found) at 62% and the fit is good in accuracy at 80% on average for the 4 tested identities, for 8 training images, and in the case of frontal faces. When the unseen face picture is best fitted by the correct class AAM, the head pose and facial deformation is retrieved. We do not need to futher decorrelate the data as it is done in [55]. We skipped the automatical initialization step of the process, just giving the manual labels as starting point for the fitting. This initialization step should be automatized later.

More extensive tests would be necessary to confirm the following rule that can be drawn from this preliminary test: we grows the training set size, and unless the face is very far from the training set, the AAM selected as winner is correct or coherent (to a group of close deformations), and the fitting is accurate.

Indeed the correctness is mainly linked to the quality of the facial database. The more care is brought to the database during its construction, the clearer the separation between close AAM classes, and the less confusion will be possible between classes for the best AAM on one test picture.

Finally, to the best of my knowledge, no existing investigation on the AAM have led to such an understanding of this method, and conducted to the well adapted solution to specialize as much as it can be each AAM of a pool of AAM helping to deal with the problem of unseen faces. The reduction of the area each model covers on the face also helps to reach higher fitting accuracy as we showed in chapter 4.

We thus paved a new way to retrieve facial deformations on unseen faces under varying poses with AAMs.

Since machine learning based solutions are efficient and well performing for the task of expression retrieval, they could be used to operate a pre-selection of the adapted AAM. Ideally, this AAM would only have to fit accurately onto the facial features. We could save the computation requirement of the long exhaustive fitting stage, retrieving the correct deformation (and pose?) through learning-based solutions, and thus accurately retrieve the shape configuration through the adapted specialized AAM.

Finally, shouldn't we be able to fasten the method of correct AAM selection with machine learning, we can always spend the time to fit the specialized AAMs exhaustively on each image, and then, from these new labels used for training, build a fast person-specific AAM that can be used to fit and track the just-learnt person.

Chapter 6

Light-Invariant AAM Fitting

6.1 Introduction

Illumination changes represent a classical problem in computer vision and many systems are unrobust to such phenomena. A lot of work is thus presented under a restricted framework where light is fixed and known.

However, human-machine interaction devices would be prone to be found anywhere. For instance, one is likely to interact with his laptop computer with embedded camera in any kind of place. The light is rarely homogeneous in the general case.

It is important to dedicate efforts to give the vision systems some robustness to light changes.

As we saw at the begining of chapter 3, the AAM is particularly unrobust to an illumination which is different from the one in the training set. It is also difficult to train an AAM with all possible illuminations. Each time a person displays a particular expression at a given pose, we should take him a picture for all possible illumination. The construction of the training database would become extremely complex. Should it be possible to do it, it would be clumsy to complexify the appearance space, when we saw from previous chapters that more specialisation of the AAM can conduct to better performances of this AAM.

In this chapter we investigate the possibility to exploit some interesting properties from the color theory to cope with illumination changes. The appearance space should not be complexified further, and we only add a step into the fitting process, where a projection of color data into a light-invariant space is performed. We introduce the concept of Light-Invariant AAM (or LI-AAM), where the AAM is now robust to illumination changes.

Under the classical optimization framework of a grey-level AAM, let's recall the form of the function to minimize:

$$\sum_{\boldsymbol{x}} \left[A_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i(\boldsymbol{x}) - I(W(\boldsymbol{x}, \boldsymbol{p})) \right]^2, \tag{6.1}$$

where:

- $A_0(\boldsymbol{x})$ represents the mean appearance image.
- $A_i(\boldsymbol{x})$ are the linear appearance bases (m dimensional: we do not consider the gain and offset component in this chapter) with coefficients $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_m]$.
- I represents the input image.
- $W(\boldsymbol{x}, \boldsymbol{p})$ represents the geometrical warp governed by parameters vector $\boldsymbol{p}$ between the image I and the reference template of same size as $A_0(\boldsymbol{x})$.

The cost function (6.1) is minimized over the warp parameters $\boldsymbol{p}$ and over the appearance coefficients $\boldsymbol{\lambda}$.

As we said, this approach is unrobust to light that is not explicitly learnt in the training set. The most common approach to address the problem of light variation is to mark shadow areas as outliers. The use of robust kernels inside the minimization process [6] allows the algorithm to reach a solution. Other approaches try to model the shadows and changes of illumination. In [7] a learning approach is used to tackle illumination changes by using a linear appearance basis. In [99] tracking under drastical illumination changes with saturation occurence is performed through an estimation of this illumination knowing the geometry of the tracked surface.

In this chapter we explore the work of Finlayson et al. in [50, 49] who introduced a shadow-invariant space into which we find it convenient to develop a new AAM fitting framework that makes the AAM robust to illumination changes. In [90], Pizarro *et al.* successfully use the shadow-invariant concept for homography registrations.

The chapter is organized as follows. In §6.2, we first present a background on color image formation, followed by the theoretical shadow-invariant image framework developed by Finlayson *et al.*

In this framework is introduced a *slope parameter* which is an internal camera parameter that needs to be estimated to obtain the shadow-invariant image.

Then in §6.3, we introduce a new cost function that projects both the training and test data into the Finlayson's shadow-invariant space, what requires to optimize on-the-fly the slope parameter together with the shape and the appearance parameters in a synthetic gauss-newton optimization scheme. Two slopes must be considered, one for the training data, and one for the input picture. We explain the different approaches that can be considered and we here develop the strategy that estimates both slope parameters online.

In §6.4 we test the developed light-invariant AAM fitting solution and compare it to the classical one.

6.2 Color image theory

6.2.1 Background on color image theory

We present the physical model used to describe the image formation process. The theory of invariant images is described later in terms and under the assumptions stated below. We consider that all the surfaces are lambertian, that the lights follow a planckian model and that the camera sensor is narrow-band. The RGB color obtained at a pixel is modeled by the following physical model:

$$\rho_k = \sigma S(\lambda_k) E(\lambda_k, T) Q_k \delta(\lambda - \lambda_k) \quad k = 1, 2, 3, \tag{6.2}$$

where $\sigma S(\lambda_k)$ represents the surface spectral reflectance functions times the lambertian factor. The term $Q_k \delta(\lambda - \lambda_k)$ represents the sensor spectral response function for each color channel k centered at wavelength λ_k. $E(\lambda_k, T)$ is the spectral power distribution of the light in the planckian model. This is modeled by the following expression:

$$E(\lambda, T) = I c_1 \lambda^{-5} \exp^{(\frac{-c_2}{T\lambda})} \tag{6.3}$$

This model holds for a high rank of color temperatures $T = [2500°, 10000°]$. The term I is a global light intensity and the constants c_1 and c_2 are fixed. According to this model, the value ρ_k obtained by the camera at any pixel is:

$$\rho_k = \sigma I c_1 \lambda_k^{-5} \exp^{\left(\frac{-c_2}{T\lambda_k}\right)} S \lambda_k Q_k \tag{6.4}$$

6.2.2 Shadow invariant image theory

The transformation allowing invariant image formation is based on the work of [7] in which a method for obtaining an illumination invariant, intrinsic image from an input color image is developed. The method relies on the above presented image formation model, based on the assumption of lambertian surfaces, narrow-band sensors and planckian illuminants. Given the three channel color components $\rho^T = (\rho_1, \rho_2, \rho_3)$ described in 6.4, the logarithm of chromaticity ratios are formed.

$$\mathcal{X}_1 = \log\left(\frac{\rho_1}{\rho_3}\right) = \log(s_1/s_3) + (e_1 - e_3)/T \tag{6.5}$$

$$\mathcal{X}_2 = \log\left(\frac{\rho_2}{\rho_3}\right) = \log(s_2/s_3) + (e_2 - e_3)/T,$$

to

(6.7)

where $e_k = -c_2/\lambda_k$ only depends on camera spectral response and not on the surface and $s_k = c_1\lambda_k^{-5}S(\lambda_k)Q_k$ does not depend on the color temperature T. The pair of values $\mathcal{X}_1$ and $\mathcal{X}_2$ lies on a line with direction vector $e = (e_1 - e_3, e_2 - e_3)$. Across different illumination temperature T, vector $\mathcal{X} = (\mathcal{X}_1, \mathcal{X}_2)$ moves along the line. An illumination invariant quantity can be formed by projecting any vector $\mathcal{X}$ onto the orthogonal line defined by $\bar{e} = (cos(\theta), sin(\theta))$. Therefore, two pixels from the same surface viewed under different illuminations get "projected" at the same place. What we care about is to make pixels from a same surface falling onto the same point on this orthogonal line defined by $\bar{e}$, but we do not care about the position onto this line, so only the parameter θ will matter for us, and no offset parameter should be considered.

The transformation $\mathcal{L}$ is simply obtained by projecting vector $\mathcal{X}$ onto the invariant line parametrized by its slope angle θ:

$$\mathcal{L}(\rho, \theta) = \mathcal{X}_1(\rho)cos(\theta) + \mathcal{X}_2(\rho)sin(\theta) \tag{6.8}$$

This transformation, as it has been previously stated, represents the mapping between a color image and its corresponding shadow invariant representation. By explicitly describing the whole color image $\mathcal{S}$ as an input in (6.8), the result of $\mathcal{L}(S, \theta)$ is a 1D shadow invariant image. The transformation is therefore global so it does not depend on pixel position $q \in \mathbb{R}$, but only on its color value.

The slope parameter θ is linked to the camera characteristics and can typically be determined by a color-chart aquired at various moments of the day while leaving unchange the camera settings. However this process may not always be possible to set up. Another solution consists in minimizing the invariant-image entropy by adjusting the slope θ (see [49]). This method can perform well only if the scene illumination is non-homogeneous. In this case, the minimum entropy over the light-invariant image is likely to be found for the θ value that projects pixels of the same material onto the same 1D light-invariant point: this is due to the fact that maximum entropy usually comes from the shadowing in the image.

In the case where illumination is homogeneous, the image major entropy does not come from light variations anymore, but rather from the difference between perceived materials in the scene, and its minimization results in a wrong θ estimation. Same material pixels are not brought to the same light-invariant point with a wrong θ parameter.

This is why this method requires the scene not to be homogeneously illuminated to allow a correct θ estimation.

In the following, we propose to integrate the shadow-invariant concept into the AAM optimization scheme and to jointly estimate the optimal parameters of shape, appearance, and θ slope.

6.3 Integrating the shadow-invariance concept into AAM fitting

In order to fit an AAM under varying lighting condition, we wish to express a new cost function to minimize that includes the shadow-invariant concept presented above. The idea is to project the training color data and the test color data into the light-invariant space, and to match them within that space. Various strategies can be set up, but some can present more advantages than others.

The formulation that would more stick to the classical one expressed by (6.1) would be the following:

$$\sum_{\boldsymbol{x}} \left[A_0^{LI}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{LI}(\boldsymbol{x}) - I^{LI}(W(\boldsymbol{x};\boldsymbol{p})) \right]^2, \tag{6.9}$$

where A_0^{LI}, all A_i^{LI} and I^{LI} are all light-invariant data, *i.e.*, data that are pre-transformed from the 3D color space to the 1D light-invariant space. In this manner, We deal like in 6.1 (where grey-level data are used) with a classical 1 dimensional set of data. This new formulation implies to compute offline the transformation of both training and test data. As we said in the previous section, this can be an ambiguous problem when the data illumination is homogeneous, what is unfortunately likely to be the case for the training data, often chosen to be homogeneously illuminated. For the test data, we wish to be able to deal with non-homogeneous as well as homogeneous light. Thus we are confronted to the same ambiguity.

A more general formulation is the following:

$$\sum_{\boldsymbol{x}} \left[\mathcal{L}\left(A_0^{C}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{C}(\boldsymbol{x}), \theta_1 \right) - \mathcal{L}(I^{C}(W(\boldsymbol{x};\boldsymbol{p}), \theta_2) \right]^2, \tag{6.10}$$

where A_0^C, A_i^C and I^C are three channel (R,G,B) color data. Within such a framework, all training and test data are converted online from color to light-invariant data. In this way, the slope associated to the training data and the slope associated to the input image can be estimated online as two extra parameters that will be added to the classical shape and appearance ones. Such a framework presents another great advantage: the ambiguity posed by the homogeneously illuminated data in the correct estimation of the θ slope is released. This is due to the following fact. Here the slope is determined together with face shape and appearance parameters of the AAM. The model appearance is compared to the image once they both have been reprojected into the light-invariant space. Whereas an entropy minimization can lead to a wrong slope θ, here a wrong slope would lead the AAM appearance to diverge from the test face image appearance, thus increasing the cost function value, which is exactly what the optimization process will tend to avoid. As a consequence, the correct slopes in the point of view of correct reprojection from color space to light-invariant space also should be the slopes that minimize Equation (6.10), independently of the nature of illumination. This particularity allows the problem to be well-posed despite the illumination homogeneity on the training data, and potentially on the test data too. In the following, will we work on the development of this formulation.

Let's now make some considerations on the following problem: it seems clear that the reprojection from test color data into the light-invariant space should benefit from the property stated above, and thus the θ slope should be estimated online, within the optimization framework. However, the θ slope relative to the training data might effectively be unknown initially, but it would be interesting for us to determine it with help of such an optimization process, and then to rely on a new cost function that directly uses the converted training data with the correct estimate of the θ slope. The use of

light-invariant training data and the search of the slope parameter associated to the input image can be formulated by the following hybrid expression:

$$\sum_{\boldsymbol{x}} \left[A_0^{LI}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{LI}(\boldsymbol{x}) - \mathcal{L}(I^C(W(\boldsymbol{x};\boldsymbol{p}), \theta_2) \right]^2 \tag{6.11}$$

We leave this interesting formulation for our future works.

Eventually, the final possibility of converting training data online, while determining offline the correct conversion of the test data can seem not to present any interest. In reality if the interest is low for an online process, it presents a great advantage to retrieve offline the slope parameter of some previously unused training data. It could be very useful to build a set of color test data that would have been particularly well characterized in θ slope (thanks to a checkerboard for instance, since it is the rightest way). This data, properly projected into light-invariant space, can serve as "Calibrating Training Data footage" to determine the training data slope in a proper manner (Of course the best would be to characterize the training data with a checkerboard too, but this might not always be possible). Let's consider the following useful formula, helping to determine the training data slope:

$$\sum_{\boldsymbol{x}} \left[\mathcal{L}\left(A_0^{C}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{C}(\boldsymbol{x}), \theta_1 \right) - I^{LI}(W(\boldsymbol{x};\boldsymbol{p})) \right]^2 \tag{6.12}$$

6.3.1 Presentation of the framework

From all the possible frameworks presented above, we decided to focus on and develop the most general one where the two slope parameters are determined online. We recall the cost function associated to this framework:

$$\sum_{\boldsymbol{x}} \left[\mathcal{L}\left(A_0^{C}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{C}(\boldsymbol{x}), \theta_1 \right) - \mathcal{L}(I^C(W(\boldsymbol{x};\boldsymbol{p}), \theta_2) \right]^2, \tag{6.13}$$

where $\theta_1 \neq \theta_2$ in the general case,*i.e.* for images taken by different cameras or even the same camera with different internal adjustments (*e.g.* white balance).

All images involved in the process are color images. To agree with dimensions in Jacobians, we suppose that given a color image I^C at pixel x, the result is a 3-component column vector.

$$I(\boldsymbol{x}) = \begin{pmatrix} I_R(\boldsymbol{x}) & I_G(\boldsymbol{x}) & I_B(\boldsymbol{x}) \end{pmatrix}^T$$

The color training set data is thus concatenated and a PCA is computed on them to build the color mean appearance A_0^C and the appearance variation components $A_i^C s$.

Under this situation we search for the minimum in (6.13) in terms of appearance coefficients $\boldsymbol{\lambda}$, warp parameters $\boldsymbol{p}$, and the angles θ_1 and θ_2 all stacked into the vector $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{p} & \boldsymbol{\lambda} & \theta_1 & \theta_2 \end{pmatrix}^T$.

As in the regular situation, a Gauss-Newton approach can easily be derived.

Expression (6.13) is modified by performing the first order Taylor expansion of the term inside brackets around parameters $\boldsymbol{v}$ to give:

$$\sum_{\boldsymbol{x}} \left[\mathcal{L}\left(A_0^{C}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{C}(\boldsymbol{x}), \theta_1 \right) - \mathcal{L}(I^C(W(\boldsymbol{x};\boldsymbol{p}), \theta_2) - J\Delta\boldsymbol{v} \right]^2, \tag{6.14}$$

where

$$J = \begin{pmatrix} \nabla\mathcal{L}(I)\frac{\partial W}{\partial p} & -\frac{\partial \mathcal{L}(S)}{\partial S}A_1 & \cdots & -\frac{\partial \mathcal{L}(S)}{\partial S}A_m & -\frac{\partial \mathcal{L}(S,\theta_1)}{\partial \theta_1} & \frac{\partial \mathcal{L}(I,\theta_2)}{\partial \theta_2} \end{pmatrix}$$

, where $S(\boldsymbol{x}) = A_0^C(\boldsymbol{x}) + \sum_{i=1}^m \lambda_i A_i^C(\boldsymbol{x})$ and

$$\Delta v = \begin{pmatrix} \Delta\boldsymbol{p} & \Delta\boldsymbol{\lambda} & \Delta\theta_1 & \Delta\theta_2 \end{pmatrix}^T$$

By taking derivatives of expression in (6.14) with respect to Δv:

$$\sum_{\boldsymbol{x}} J^T \left[\mathcal{L}\left(A_0^C(\boldsymbol{x}) + \sum_{i=1}^m \lambda_i A_i^C(\boldsymbol{x}), \theta_1 \right) - \mathcal{L}(I^C(W(\boldsymbol{x};\boldsymbol{p}),\theta_2) - J\Delta\boldsymbol{v} \right], \tag{6.15}$$

We denote:

$$SD = J \tag{6.16}$$

By setting the expression in (6.15) to equal zero and solving we get the typical one step iteration of a Gauss-Newton approximation.

$$\Delta\boldsymbol{v} = H^{-1} \sum_{\boldsymbol{x}} SD^T(\boldsymbol{x}) E(\boldsymbol{x}), \tag{6.17}$$

where H is the Gauss-Newton approximation of the Hessian,

$$H = \sum_{\boldsymbol{x}} SD^T(\boldsymbol{x}) SD(\boldsymbol{x})$$

and E is the error image defined as

$$E(\boldsymbol{x}) = \mathcal{L}\left(A_0^C(\boldsymbol{x}) + \sum_{i=1}^m \lambda_i A_i^C(\boldsymbol{x}), \theta_1 \right) - \mathcal{L}(I^C(W(\boldsymbol{x};\boldsymbol{p}),\theta_2)$$

.

The problem of the presented approach comes from the fact that some of the derivatives involved in the computation of the Jacobian J are non linear, especially for the appearance parameters. As an example:

$$\frac{\partial \mathcal{L}(S)}{\partial S} = \left(\frac{\partial \mathcal{L}(S)}{\partial S_R}, \frac{\partial \mathcal{L}(S)}{\partial S_B}, \frac{\partial \mathcal{L}(S)}{\partial S_G} \right) = \left(\frac{1}{S_R} cos(\theta), \frac{1}{S_B} sin(\theta), \frac{-1}{S_G} cos(\theta) + \frac{-1}{S_G} sin(\theta) \right)$$

An advantage of the approach is that we can use the RGB color input data without operating any modification on it. However a small pre-process change on the input data can lead to nice improvements in the mathematical developments.

In the following we develop the framework under various mathematical tricks. In §6.3.2 the log-Chromaticity space is introduced, leading to great simplification in the Jacobian computation. The RGB data, 3 dimensional, is pre-converted and sent to the 2 dimensional log-Chromaticity space. This is probably the development of the LI-AAM that provides the best performances. Thus we consider it useful to decline the forward additive version into the inverse compositional version of the LI-AAM in §6.3.3. The drawback of this solution is that the data conversion from the RGB to the log-Chromaticity space imply to lose information, what makes impossible the step backward, allowing to recover the color information.

Despite all this development under the log-Chromaticity solution, this is not the algorithm we will test, but instead we rely on another mathematical development of the optimization framework. In

§6.3.4 we introduce the log-RGB space where the Jacobian also has a simple form. The performance we can expect from the log-RGB's framework is probably slightly smaller than those possibly obtained with the log-Chromaticity's one (this is why it is interesting to present its formulation). However we bet on the development of the log-RGB's framework for one reason: there is no loss involved in the conversion of the data from RGB to log-RGB, both 3 dimensional spaces. This is interesting because after fitting the face under any illumination context, we plan for a future work, to recover the equivalent facial appearance with an homogeneous illumination. Indeed, this can be a useful feature for facial recognition devices that are usually unrobust to unknown illumination for instance.

6.3.2 Linear appearance in Log-Chromaticity space

A simpler approach can be obtained if we assume that we have a linear appearance basis directly in $\log(R/G), log(B/G)$ coordinates.

By converting all training images from RGB to the log-Chromaticity space, a new appearance basis is obtained by applying PCA to the set of images.

$$Ac_i(\boldsymbol{x}) = \begin{pmatrix} Ac_i(\boldsymbol{x})_{R/G} & Ac_i(\boldsymbol{x})_{B/G} \end{pmatrix}^T \quad i = 1, \cdots, m$$

and

$$Ac_0(\boldsymbol{x}) = \begin{pmatrix} Ac_0(\boldsymbol{x})_{R/G} & Ac_0(\boldsymbol{x})_{B/G} \end{pmatrix}^T$$

Obtaining the shadow-invariant image given any image already in log-Chromaticity space is a linear transformation, so we define $\mathcal{L}(Ac_0(x), \theta)$ with input image in log-Chromaticity space as follows:

$$\mathcal{L}(Ac_0(\boldsymbol{x}), \theta) = L(\theta) \cdot Ac_0(\boldsymbol{x}) = Ac_0(\boldsymbol{x})_{R/G} cos(\theta) + Ac_0(\boldsymbol{x})_{B/G} sin(\theta), \tag{6.18}$$

where $L(\theta) = \begin{pmatrix} cos(\theta) & sin(\theta) \end{pmatrix}$

By using appearance basis in log-chromaticity space, the new cost function to minimize is:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) Ac_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i L(\theta_1) Ac_i(\boldsymbol{x}) - \mathcal{L}(I(W(\boldsymbol{x}; \boldsymbol{p}), \theta_2) \right]^2, \tag{6.19}$$

Under this transformation Jacobians are greatly simplified with the appearance terms:

$$\frac{\partial \mathcal{L}(Sc)}{\partial Sc} = \left(\frac{\partial \mathcal{L}(Sc)}{\partial Sc_{R/G}}, \frac{\partial \mathcal{L}(Sc)}{\partial Sc_{B/G}} \right) = (cos(\theta), sin(\theta)) = L(\theta)$$

where $Sc(\boldsymbol{x}) = Ac_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i Ac_i(\boldsymbol{x})$

The resulting jacobian is:

$$J = \begin{pmatrix} \nabla \mathcal{L}(I) \frac{\partial W}{\partial p} & -L(\theta_1) Ac_1 & \cdots & -L(\theta_1) Ac_m & -\frac{\partial \mathcal{L}(Sc, \theta_1)}{\partial \theta_1} & \frac{\partial \mathcal{L}(I, \theta_2)}{\partial \theta_2} \end{pmatrix}$$

,

Each term $L(\theta_1) Ac_i$ is directly the shadow invariant image of each basis element Ac_i.

The terms $\frac{\partial \mathcal{L}(Sc, \theta_1)}{\partial \theta_1}$ are also linear:

$$\frac{\partial \mathcal{L}(S, \theta_1)}{\partial \theta_1} = L(\theta_1)^{\perp} S \tag{6.20}$$

where $L(\theta_1)^{\perp} = (-sin(\theta_1), cos(\theta_1))$ and $L^{\perp} L^T = 0$, so they are the projection of the log-Chromaticity space in the orthogonal line to the one defined by θ_1.

6.3.3 Adaptation to the Simultaneous Inverse Compositional

We now want to adapt the approach to the Simultaneous Inverse Compositional proposed in [9]

The warp is now applied to the source image composed by the template and the apprance basis to obtain an increment which will be inverted to obtain the equivalent incremental warp in the input image I. The cost function is the following:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) Ac_0(W(\boldsymbol{x};\Delta\boldsymbol{p})) + \sum_{i=1}^{m} \lambda_i L(\theta_1) Ac_i(W(\boldsymbol{x};\Delta\boldsymbol{p})) - \mathcal{L}(I(W(\boldsymbol{x};\boldsymbol{p}),\theta_2) \right]^2, \tag{6.21}$$

We minimize simultaneously around $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{p} & \boldsymbol{\lambda} & \theta_1 & \theta_2 \end{pmatrix}^T$, and then we update the warp in inverse compositional form and the rest in forward additive:

$$W(\boldsymbol{x};\boldsymbol{p}) \leftarrow W(\boldsymbol{x};\boldsymbol{p}) \circ W(\boldsymbol{x};\Delta\boldsymbol{p})^{-1} \tag{6.22}$$

$$\lambda_i \leftarrow \lambda_i \Delta\lambda_i \quad \theta_i \leftarrow \theta_i + \Delta\theta_i \tag{6.23}$$

$Ac_i(W(\boldsymbol{x};\Delta\boldsymbol{p}))$ and assuming that $W(\boldsymbol{x};0)$ is the identity warp, we get:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) \left((Ac_0(\boldsymbol{x}) + \nabla Tc \frac{\partial W}{\partial p})\Delta\boldsymbol{p} + \sum_{i=1}^{m} (\lambda_i + \Delta\lambda_i)(Ac_i(\boldsymbol{x}) + \nabla Ac_i \frac{\partial W}{\partial p}\Delta\boldsymbol{p}) \right) - \mathcal{L}(I(W(\boldsymbol{x};\boldsymbol{p}),\theta_2) \right]^2,$$

Neglecting second order terms the resulting function is:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) \left(Ac_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i Ac_i(W(\boldsymbol{x};\Delta\boldsymbol{p})) \left(\nabla Ac_0 + \sum_{i=1}^{m} \lambda_i \nabla Ac_i \right) \frac{\partial W}{\partial p}\Delta\boldsymbol{p} \right) - \mathcal{L}(I(W(\boldsymbol{x};\boldsymbol{p}),\theta_2) \right]^2,$$

The Jacobian of the term inside brackets is:

$$J(\boldsymbol{x}) = SD(\boldsymbol{x}) = \left(L(\theta_1) \left(\nabla Sc(\boldsymbol{x}) \frac{\partial W}{\partial p}, A_1, \cdots, A_m \right), L(\theta_1)^{\perp} Sc(\boldsymbol{x}), -\frac{\partial \mathcal{L}(I(W(\boldsymbol{x};\boldsymbol{p})),\theta_2)}{\partial \theta_2} \right) \tag{6.24}$$

where

$$Sc(\boldsymbol{x}) = Ac_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i Ac_i(\boldsymbol{x})$$

and

$$\nabla Sc(\boldsymbol{x}) = \nabla Ac_0(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \nabla Ac_i(\boldsymbol{x})$$

Finally the so called error image $E(\boldsymbol{x})$:

$$E(\boldsymbol{x}) = L(\theta_1) Sc(\boldsymbol{x}) - \mathcal{L}(I(W(\boldsymbol{x};\boldsymbol{p}),\theta_2) \tag{6.25}$$

The Gauss-Newton update is:

$$\Delta\boldsymbol{v} = -H^{-1} \sum_{\boldsymbol{x}} SD(\boldsymbol{x})^T E(\boldsymbol{x}) \tag{6.26}$$

where H^{-1} is the inverse of the (Gauss-Newton approximation of the) Hessian:

$$H = \sum_{\boldsymbol{x}} SD^T(\boldsymbol{x}) SD(\boldsymbol{x}) \tag{6.27}$$

As was commented before we solve iteratively in Δv and the composition for angles and coefficients is forward additive and the warp is inverse compositional.

The algorithm is not efficient because the Hessian is changing at each iteration, so its inverse must be calculated which increases the complexity.

We now present another simplified version of the LI-AAM optimization process. It is a forward additive version that we will be used for the tests. (the inverse compositional presents few interest since it is not efficient and the gradient must be recomputed at each iteration).

6.3.4 Linear appearance in Log-RGB space

Another simpler approach is proposed based on a linear appearance basis expressed in the

$$(\log(R), \log(B), \log(G))$$

coordinates. The transformation is reversible and continuous assuming strictly positive image coordinates. This will be useful in the case we want to step backward in the process and retrieve an estimation of the shadow-free image corresponding to a processed image (under non-homogeneous illumination).

By converting all training images from RGB to the log-RGB space, a new appearance basis is obtained by applying PCA to the set of images:

$$A_i^{log}(\boldsymbol{x}) = (A_i^{log}(\boldsymbol{x})_R, A_i^{log}(\boldsymbol{x})_B, A_i^{log}(\boldsymbol{x})_G)^T \quad i = 1, \cdots, m$$

and

$$A_0^{log}(\boldsymbol{x}) = (A_0^{log}(\boldsymbol{x})_R, A_0^{log}(\boldsymbol{x})_B, A_0^{log}(\boldsymbol{x})_G)^T$$

Obtaining the shadow invariant image $\mathcal{L}$ given any image already in Log-RGB space is a linear transformation, so we define $\mathcal{L}(I^{log}(\boldsymbol{x}), \theta)$ with input image I^{log} in log-RGB space as follows:

$$\mathcal{L}(I^{log}(\boldsymbol{x}), \theta) = L(\theta).I^{log}(\boldsymbol{x}), \tag{6.28}$$

where $L(\theta) = (cos(\theta), sin(\theta), -sin(\theta) - cos(\theta))$

We can therefore write the new cost function in the following way:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) A_0^{log}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i L(\theta_1) A_i^{log}(\boldsymbol{x}) - L(\theta_2) I^{log}(W(\boldsymbol{x}; \boldsymbol{p})) \right]^2, \tag{6.29}$$

that can also be written

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) S^{log}(\boldsymbol{x}) - L(\theta_2) I^{log}(W(\boldsymbol{x}; \boldsymbol{p})) \right]^2, \tag{6.30}$$

for which $S^{log} = A_0^{log}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{log}(\boldsymbol{x})$

We search for the minimum in function of vector $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{p} & \boldsymbol{\lambda} & \theta_1 & \theta_2 \end{pmatrix}^T$.

Under this transformation Jacobians are greatly simplified with the appearance terms:

$$\frac{\partial \mathcal{L}(S^{log}, \theta_1)}{\partial S^{log}} = \left(\frac{\partial \mathcal{L}(S^{log}, \theta_1)}{\partial S_R^{log}}, \frac{\partial \mathcal{L}(S^{log}, \theta_1)}{\partial S_B^{log}}, \frac{\partial \mathcal{L}(S^{log}, \theta_1)}{\partial S_G^{log}} \right) = L(\theta_1)$$

The resulting jacobian is:

$$J = \left(\nabla L(\theta_2) I^{log} \frac{\partial W}{\partial p}, -L(\theta_1) A_1^{log}, \cdots, -L(\theta_1) A_m^{log}, -\frac{\partial L(\theta_1) S^{log}}{\partial \theta_1}, \frac{\partial L(\theta_2) I^{log}(W(\boldsymbol{x};\boldsymbol{p}))}{\partial \theta_2} \right),$$

Each term $L(\theta_1)A_i^{log}$ is directly the shadow invariant image of each basis element A_i^{log}. $\nabla L(\theta_2)I^{log}$ is the gradient of the invariant representation of the input image I^{log} warped with the current parameter p, and we have:

$$\frac{\partial L(\theta_2) I^{log}(W(\boldsymbol{x};\boldsymbol{p}))}{\partial \theta_2} = L(\theta_2)^{\perp} I^{log}(W(\boldsymbol{x};\boldsymbol{p})), \tag{6.31}$$

and

$$\frac{\partial L(\theta_1) S^{log}}{\partial \theta_1} = L(\theta_1)^{\perp} S^{log} \tag{6.32}$$

Expression 6.29 is modified by performing the first order Taylor expansion of the term inside brackets around the parameters grouped in vector v to give:

$$\sum_{\boldsymbol{x}} \left[L(\theta_1) \left(A_0^{log}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{log}(x) \right) - L(\theta_2) I^{log}(W(x;p)) + J\Delta v \right]^2, \tag{6.33}$$

The minimum of (6.33) with respect to increment Δv is obtained by taking derivatives:

$$\sum_{\boldsymbol{x}} J^T \left[L(\theta_1) \left(A_0^{log}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{log}(x) \right) - L(\theta_2) I^{log}(W(x;p)) + J\Delta v \right], \tag{6.34}$$

We denote:

$$SD = J \tag{6.35}$$

By setting the expression (6.34) to zero and solving:

$$\Delta p = H^{-1} \sum_{x} SD^T(\boldsymbol{x}) E(\boldsymbol{x}), \tag{6.36}$$

where H is the matrix approximation of the Hessian,

$$H = \sum_{x} SD^T(\boldsymbol{x}) SD(\boldsymbol{x})$$

and $E(\boldsymbol{x})$ is the error image

$$E(\boldsymbol{x}) = L(\theta_1) \left(A_0^{log}(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i A_i^{log}(\boldsymbol{x}) \right) - L(\theta_2) I^{log}(W(x;p))$$

6.4 Experimental evaluation

The following tests are set up with the forward additive version of the LI-AAM fitting algorithm described in 6.3.4. The algorithm optimizes online both the training slope θ_1 and the test image slope θ_2.

6.4.1 Tracking test under illumination variations

In this test, we took two rafales of pictures with a canon reflex camera. In the first scene, a person displaying a neutral expression performs a head rotation laterally from frontal to left, to frontal, to right, to frontal again, and ends performing a smile. The illumination is homogeneous in this first scenario. In the second sequence, the same person performs the same moves, but under a lateral illumination.

We manual label the key-frames of the homogeneously illuminated sequence of images. These frames will be used for training both a classical-AAM and a LI-AAM. For both AAMs we retain 90% shape and appearance variance.

Sequence tracking. The test is performed on the laterally illuminated sequence. We observe the tracking behaviour of both AAMs on this sequence. The classical tracking is clearly non-robust to illumination changes and diverges from the very first frame. The Light Invariant tracking shows to cope perfectly with this unknow illumination.

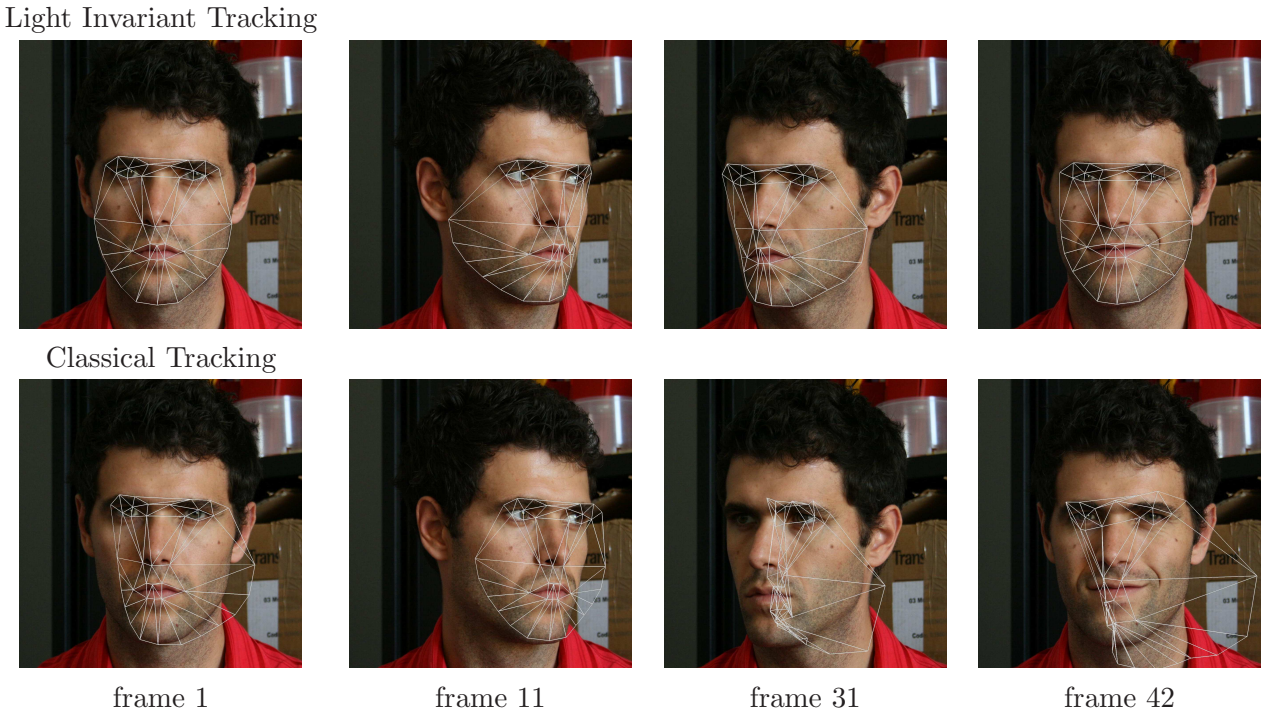


Figure 6.1: Light Invariant and Classical tracking are compared on a sequence of pictures representing a face laterally illuminated. The training pictures used to train both AAMs are extracted from a sequence in all similar to the test sequence, excepted in that illumination is homogeneous. The Classical tracking is clearly non-robust to illumination changes, whereas the light invariant solution shows to perform impressively well under unknown illumination.

Some extra fitting results on pictures presenting other illuminations. The LI-AAM fitting process is initialized close to the solution given by manual labels. It stays into initial position, whereas the classical-AAM would diverge immediately.

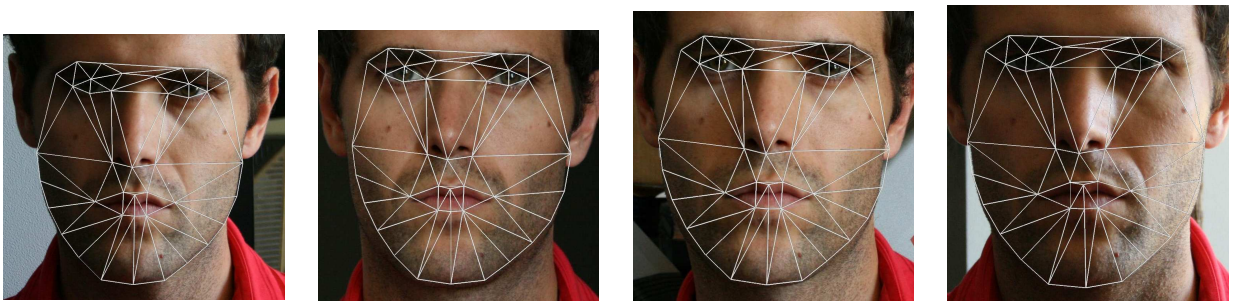


Figure 6.2: Some extra fitting results under different kind of illumination. The LI-AAM shows to be robust to any non-saturated contexts.

Frequence of Convergence. The frequence of convergence is tested and we compare the robustness to initial perturbation of the LI-AAM against the robustness of the classical-AAM (using inverse compositional fitting algorithm) on the first frame of the laterally illuminated sequence. The LI-AAM is trained with the key-frames relative to the homogeneously illuminated sequence. We already saw that the classical-AAM completely fails to fit the laterally illuminated face when it is trained with homogeneously illuminated face frames. Then, when train the classical-AAM the key-frames relative to the laterally illuminated sequence. It is clearly favorable for the classical-AAM in this context where what is fitted is what is learnt. However, this experiment provides a way to benchmark the LI-AAM performance against the classical-AAM's.

Figure 6.3 presents the percentage of convergence on 20 trials for each perturbation intensity we tested. We randomly generated similarity transform perturbations on the ground-truth shape, and used the average point to point error to select the initial positions we needed for the test. We chose to do 20 trials for each perturbation intensity tested. The range of perturbation intensities varies between 10 and 110 point to point error in pixels, with a step of 10. Convergence is claimed when the final fitting point to point error after 60 iterations is lower than 12 pixels. The model is close enough from the solution at this distance and we can suppose it has converged.

On figure 6.4 are illustrated four examples of initial positions on the test frame for different initial perturbation intensities.

6.5 Conclusion

In this chapter we present the concept of Light-Invariant AAM or LI-AAM. Among many possible declinations that we present, one framework is developed to face the problem of light changes on faces while fitting an AAM. The solution we propose aims to minimize a new cost function that involves the estimation of two slope parameters together with the shape and appearance parameters. Those slope parameters depend on the camera, and the framework we develop allows the use of different cameras for building the training set and taking the test images.

We test our light-invariant fitting strategy on a sequence where the illumination is different from the one saw on training pictures, and we compare it to the classical AAM fitting algorithm. The results we obtain exceed our expectances, and the method shows to be reliable. The frequence of convergence shows that the solution is less robust to initial perturbations than the classical one, which is due to the higher complexity of the optimization process that deals with two extra parameters expressing new physical phenomena in the function to minimize.

In future work, we plan to develop the strategy that easily calibrate any training data from a simple

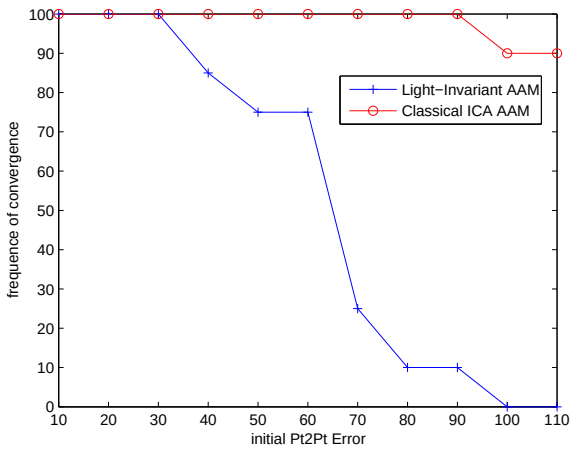
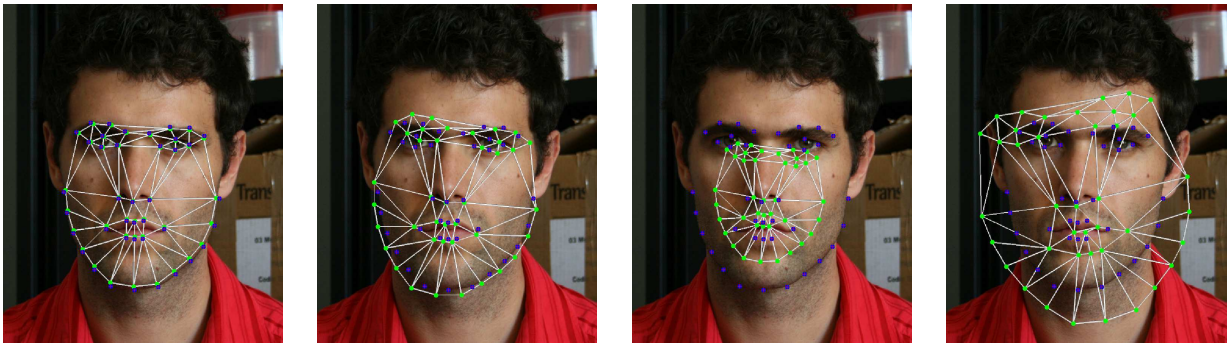


Figure 6.3: Frequences of convergence of the LI-AAM and of the classical-AAM on the first frame of the laterally illuminated sequence for various initial perturbation intensity measured in pixel average point to point error among all model vertices position against the ground-truth shape.



(a) (b) (c) (d)

Figure 6.4: Examples of initial perturbations of the ground-truth shape on the first frame of the laterally illuminated sequence. The ground-truth shape is represented with red crossed blu dots. The perturbed model position is showed with the mesh-wired model. One example for four different perturbation intensities is represented: in (a), an example of 10 pixel, in (b), 40 pixel, in (c), 80 pixel and in (d), 110 pixel point to point perturbation are represented.

pre-calibrated light-invariant footage. The training data can thus be converted into light-invariant images offline, and one slope does not need to be computed anymore. Only the input picture's slope parameter should be estimated online. This should also result in a robustness improvement of the light-invariant fitting.

We are investigating the possibility to reconstruct the appearance without shadow. This can find applications in people authentication systems for example.

A natural question to answer is, can the light-invariant AAM method be easily used to deal with unseen faces? This is what we will answer in a future work too.

Chapter 7

Segmentation and Substitution of Facial Artefacts

AAMs are usually used to fit and track faces on video sequences. But the generative power of an AAM can be used to different purposes too. Interesting applications can be built upon this AAM generability.

In this chapter we entertainingly investigate the possibility to use the AAM appearance component to segment particular artefacts on the face, like facial hair, and glasses, and also to generate the equivalent free-from-artefact face.

The proposed solution could be for example integrated to a face recognition system, usually not designed to deal with glasses, or maybe not knowing the person with facial hair. Other applications can for instance be found in photo enhancement, relooking, glasses virtual trying.

This work was presented in [88].

7.1 Introduction

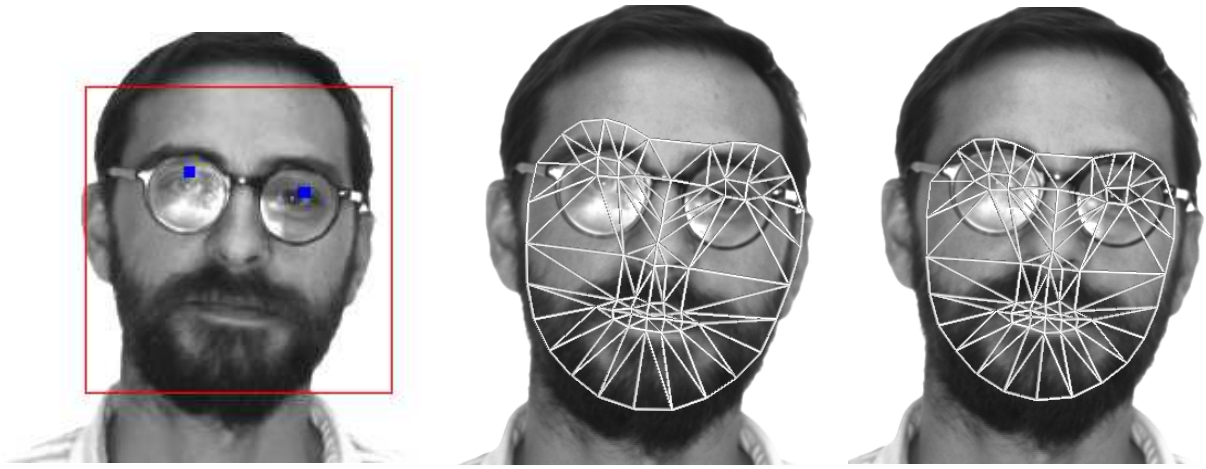
Many facial analysis solutions are proposed to deal with faces that do not wear glasses. In the presence of glasses, these solutions' performances can decrease dramatically. In a context of human-machine interface, the user could virtually try new pairs of glasses, but should he take off his own pair, preventing him from watching the screen correctly? It would be interesting to virtually remove the glasses and replace them with *non-glasses* pixels before applying a new virtual pair on the face.

Being a rigid object, glasses on the face can also advantageously be used to accurately track the face pose following a non-deformable structure. But the glasses should be detected if we want to opportunely trigger a special glasses tracker solution.

Glasses detection and substitution with non-glasses pixels is therefore very useful. For other applications like relooking, it can be interesting to segment and substitute other artefacts like facial hair for example.

Some works on glasses removal were proposed. Wu *et al.* [109] bound the glasses area through a Markov Chain, and entirely substitute this area with an equivalent free-from-glasses one, estimated through reprojection within an eigenspace. It is however unnecessary to substitute all pixels, especially since the eigenspace filters the data that can appear blurred at rendering. Park et al. [85] more thoroughly estimate glasses pixels and aim to substitute those only. But the substituting pixels are estimated from eigenfaces on which the semantical matching between pixels is not always ensured, what can result in a blurred reconstructed area.

Our approach is exclusively based on a couple of AAM appearance components to detect the glasses



(a) (b) (c)

Figure 7.1: (a) eye center detection, (b) AAM initialisation, and (c) converged AAM position.

and facial hair pixels. We strive to segment those pixels only in order to substitute the minimum amount of pixels and to degrade as less as possible the original face on the picture. The same AAM components are used to statistically estimate the non-hair and non-glasses equivalent pixels to be placed on the face.

A generic global AAM trained to fit on frontal and neutral faces is fitted onto an unseen face: the eye center detector allows to initialize the model on the face, and its position is thus optimized to accurately fit the face. The model is here built on 60 face images: the 40 used in chapter 4, and 20 others representing women faces all extracted from the AR database. The accuracy obtained with the global model is in general sufficient for this application. The extra accuracy brought by our segmented models are not necessary here (at least for this first version of the solution). We illustrate this initialization and fitting process on figure 7.1.

Once the model positionned on the image, its semantical information becomes available and we can start studying this face.

In §7.2 we present the way to detect the hair and glasses areas and we explain how we substitute them with non-hair and non-glasses pixels. Finally in §7.3 we present some results obtained with the proposed method, and we conclude and give perspectives in §7.4.

7.2 Artefact segmentation and substitution strategy

From the model position on the face image, we extract and project the facial appearance onto the mean face shape to normalize the appearance size and shape as illustrates the Figure 7.2. We compute this projection by use of the triangle to triangle matching explained in §2.1.2. Once shape normalized, this appearance is expressed by a vector $\boldsymbol{g}_{extr}$ of appearance describing le gray level of each pixel of the face.

The employed strategy aims to reconstruct the vector $\boldsymbol{g}_{extr}$ with help of an appearance basis B_h built from images free from facial hair. The reconstructed appearance is $\boldsymbol{g}_h$. The difference $|\boldsymbol{g}_{extr} - \boldsymbol{g}_h|$ should indicate with higher error the hair area. Equivalently for glasses extraction, we create a B_g basis built from faces not wearing glasses with which we attempt to express $\boldsymbol{g}_{extr}$, then obtaining the reconstructed appearance $\boldsymbol{g}_g$. The difference $|\boldsymbol{g}_{extr} - \boldsymbol{g}_g|$ should here indicate with higher error the glasses area.

$\boldsymbol{g}_{extr}$

Figure 7.2: Facial appearance extraction through projection onto the mean face shape $\boldsymbol{s}_0$ of the pixels lying under the AAM represented on figure 7.1 (c).

To build the basis B_h, we retained the faces not presenting facial hair (23 in total) among the 60 faces we extracted from the AR database. We reproject the appearances of these faces onto the mean shape $\boldsymbol{s}_0$ of the generic global model built on all 60 images, and a PCA is applied on the appearance data to retrieve a number of eigencomponents that express 95% of the total appearance variance of the faces with no facial hair, which represents 18 components. To these components we add the two extra offset and gain components $\boldsymbol{g}_{h_1}$, vectors of 1s that we normalize, and $\boldsymbol{g}_{h_0}$, average of all training appearances. We obtain the basis $B_h = [\boldsymbol{g}_{h_0}, \boldsymbol{g}_{h_1}, \cdots, \boldsymbol{g}_{h_m}]$.

In the same fashion we build the basis B_g, selecting the training images of faces without glasses (47 images among 60). A PCA applied on the free-from-glasses data gives us 36 eigencomponents to express 95% of the total appearance variance contained in the training database. We add the offset and gain components here too to obtain the basis $B_g = [\boldsymbol{g}_{g_0}, \boldsymbol{g}_{g_1}, \cdots, \boldsymbol{g}_{g_n}]$.

Let's now detail the process that extract the facial hair on $\boldsymbol{g}_{extr}$. A similar process is employed to extract the glasses. An optimization is performed to express the vector $\boldsymbol{g}_{extr}$ as a linear combination of the components of the basis B_h. We look for the vector of parameters $\boldsymbol{p}_h$ that minimizes the sum over all pixels x of the following square differences

$$\sum_x [\boldsymbol{g}_{extr}(x) - B_h^x \boldsymbol{p}_h]^2 \tag{7.1}$$

where B_h^x is the xth row of matrix B_h. We obtain $\boldsymbol{g}_h = B_h \boldsymbol{p}_h$, result of the reconstruction that means to represent the person's face without his facial hair, if any. In reality, this simple reconstruction suffers from the following problem: when the person effectively has facial hair, the hair area on $\boldsymbol{g}_{extr}$ introduces a bias on the mean gray-level of $\boldsymbol{g}_h$ that tends to get close to the gray-level of the hair, when at the contrary, we wish to obtain a frank difference between $\boldsymbol{g}_h$ and $\boldsymbol{g}_{extr}$ over the hair area to permit a better segmentation.

The strategy we adopt to solve the problem consists to minimize a weighted sum of square differences

$$\sum_x \boldsymbol{Q}(x).\,[\boldsymbol{g}_{extr}(x) - B_h^x \boldsymbol{p}_h]^2 \tag{7.2}$$

where $\boldsymbol{Q}$ is a vector of weights giving a certain level of consideration during the minimization: pixels within the potential facial hair area are less considerated than other pixels. In our test we give a null weight to those potentially hair pixels. This process is illustrated on figure 7.3. The minimization of the equation (7.2) has a closed-form solution that is detailled in annexe 7.4.

Pixels of $\boldsymbol{g}_{extr}$ that are considered | $\boldsymbol{g}_h$ | I_h

Figure 7.3: The only pixels that we consider from the original appearance are those lying out of the potential hair area. These pixels are used to search for $\boldsymbol{g}_h$, the estimation of this appearance without facial hair. I_h is the absolute difference between $\boldsymbol{g}_{extr}$ and its reconstruction without hair $\boldsymbol{g}_h$.

Relevant pixels on I_h | Segmentation (error > 40) | Corresponding pixels on the original

Figure 7.4: The potential facial hair pixel map is cut to conserve the area concerned by this artefact only. It can therefore be thresholded as we do it here in order to segment binarily the hair pixels. These pixels are reprojected (and indicated in red) onto the original image.

Once the reconstruction $\boldsymbol{g}_h$ is obtained, the difference $I_h = |\boldsymbol{g}_{extr} - \boldsymbol{g}_h|$ is the representative map of the facial hair area of this person. This map can be used as confidence map for a recognition system, or for tracking for instance. We can also binarize it (determining a threshold) to segment the hair area on the image. In the solution we propose here, we perform a binarization: we retain as facial hair all pixels exceeding an error of 40 on I_h, and we inversely reproject the segmented pixel position onto the original image as illustrates it the figure 7.4). We empirically chose the threshold, but we observed that good results can be obtained for any value between 30 and 50.

We then try to substitute the hair pixels with non-hair pixels. To this end we take advantage of the generative power of the AAMs. The idea consists to determine what would be the gray-level value of each segmented pixel by only watching at the non-segmented ones that are close to the facial hair.

A similar process is applied for the glasses. We look for the vector $\boldsymbol{p}_g$ minimizing the following

Segmentation (error <= 40) | appearance without hair | Insertion of the non-hair pixels

Figure 7.5: Within the hair area, pixels that are segmented *non-hair* (white pixels over the left image) are considered to evaluate the appearance instance without hair (central image) closest from these pixels. The pixels that were segmented as hair can consequently be replaced by their non-hair estimate taken on this instance. These pixels are therefore reprojected onto the original image (right image).

equation:

$$\sum_x \boldsymbol{Q}(x).\left[\boldsymbol{g}_{extr}(x) - B_g^x \boldsymbol{p}_g\right]^2 \tag{7.3}$$

where B_g^x is the xth row of matrice B_g. We obtain $\boldsymbol{g}_g = B_g \boldsymbol{p}_g$, result of the estimated reconstruction of the person's face without his/her glasses, if he/she wear a pair. The vector of weights $\boldsymbol{Q}$ is set to 1 where the pixel is located outside of the potential glasses area, and to 0 inside of this area. The segmenting process of the glasses is illustrated on the first two rows of the figure 7.6.

We also substitute the glasses pixels with non-glasses pixels. This time we dilate the obtained segmentation to be sure to properly cover all the glasses area: a dilation is applied onto the binary image relative to the glasses segmentation with an amplitude that is a function of this image support size. Non-glasses segmented pixels are looked at to search for the non-glasses appearance instance the closest from glasses pixels. The glasses pixels are replaced in the original image with their non-glasses equivalent as illustrates it the figure 7.6, on last row. The eyes, when they are claimed to be glasses by the segmentation process, are not substituted on the original image: we consider that this area could not be well reconstructed by the basis B_g since the eyes appearance cannot a priori be induced just watching the rest of the face. We thus prefer to conserve them the way they are on the original image, even if a specularity can be seen on it. A post-treatment could be applied to reduce this specular effect. We protect the eye area from the substitution since we semantically know where it is located under the AAM we initially fitted on that face.

7.3 Results

We here show some of the representative results we obtained with the method presented above on figure 7.7. Segmentation is of very good quality for facial hair. For glasses, we make use of a dilation of the segmented area to prevent some glasses pixels not to be substitute on the original image. The results we obtain are good in quality and glasses are almost everywhere correctly replaced by non-glasses pixels. Some small face areas that are non-glasses areas can sometimes be segmented as glasses. The

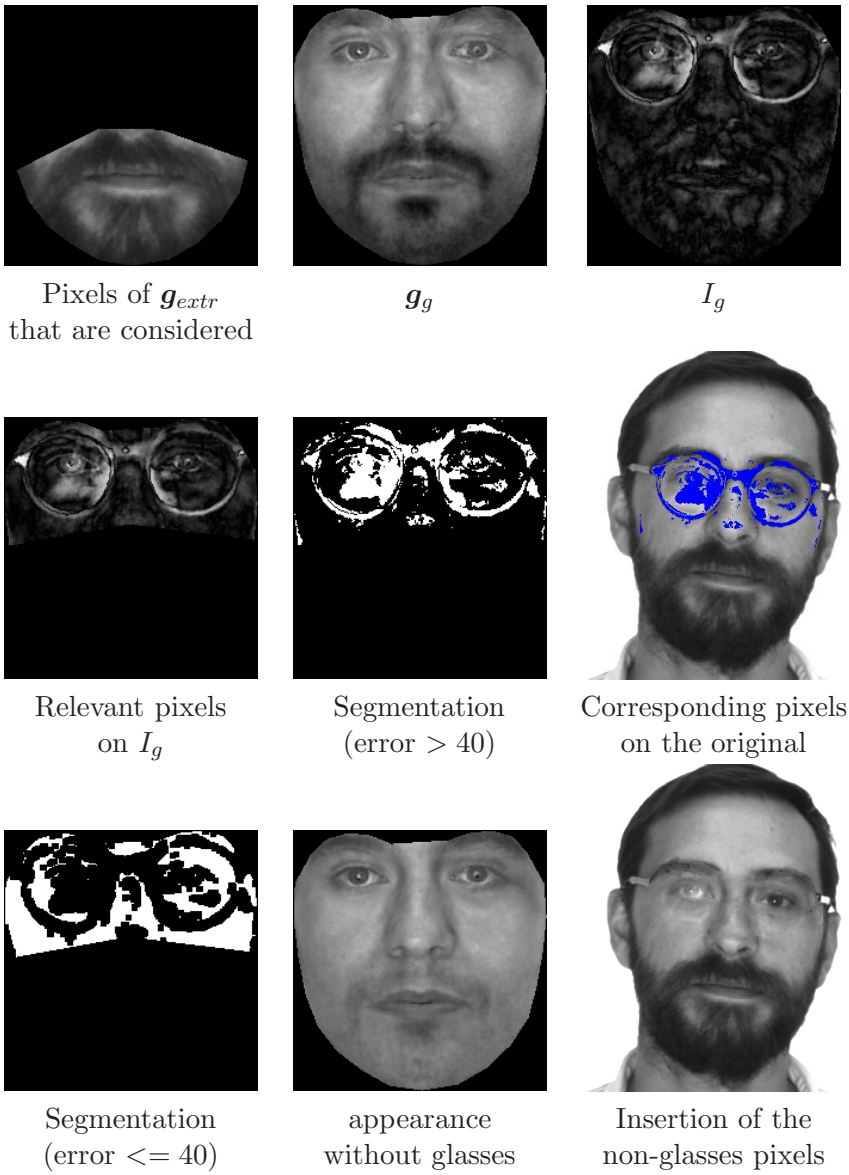


Figure 7.6: Illustration of the glasses segmentation and substitution by their non-glasses corresponding pixels.

solution should be enhanced with an extra process able to analyse the segmented areas and determine where the glasses really are. False positives could therefore be eliminated and the glasses could more accurately be described.

7.4 Conclusion

In this chapter we present a method that makes a special use of the AAM appearance component. The construction of appearance eigenspaces free from facial hair or free from glasses allows to segment facial areas that contain such artefacts. Once they are segmented we show how it is possible to substitute these artefact pixels with their equivalent non-artefact ones with the help of face pixels that are

declared as non-artefact at segmentation. The method globally provides good results. However the segmentation can lead to some small false positive and false negative segmented areas. For the case of glasses segmentation, this could be solved by adding some a priori information: a fine glasses analysis and a search for a best match among a bank of possible glasses could be developed to match the glasses in the input picture, then refining the segmentation further and suppress false positive.

Another improvement could be obtained with the use of color: an AAM built on the three color channels (as we did in chapter 6) could bring more robustness to the glasses segmentation since this item usually clearly get into color contrast with the skin on the face. Working with color would also allow to substitute the segmented pixels in color on an original color image.

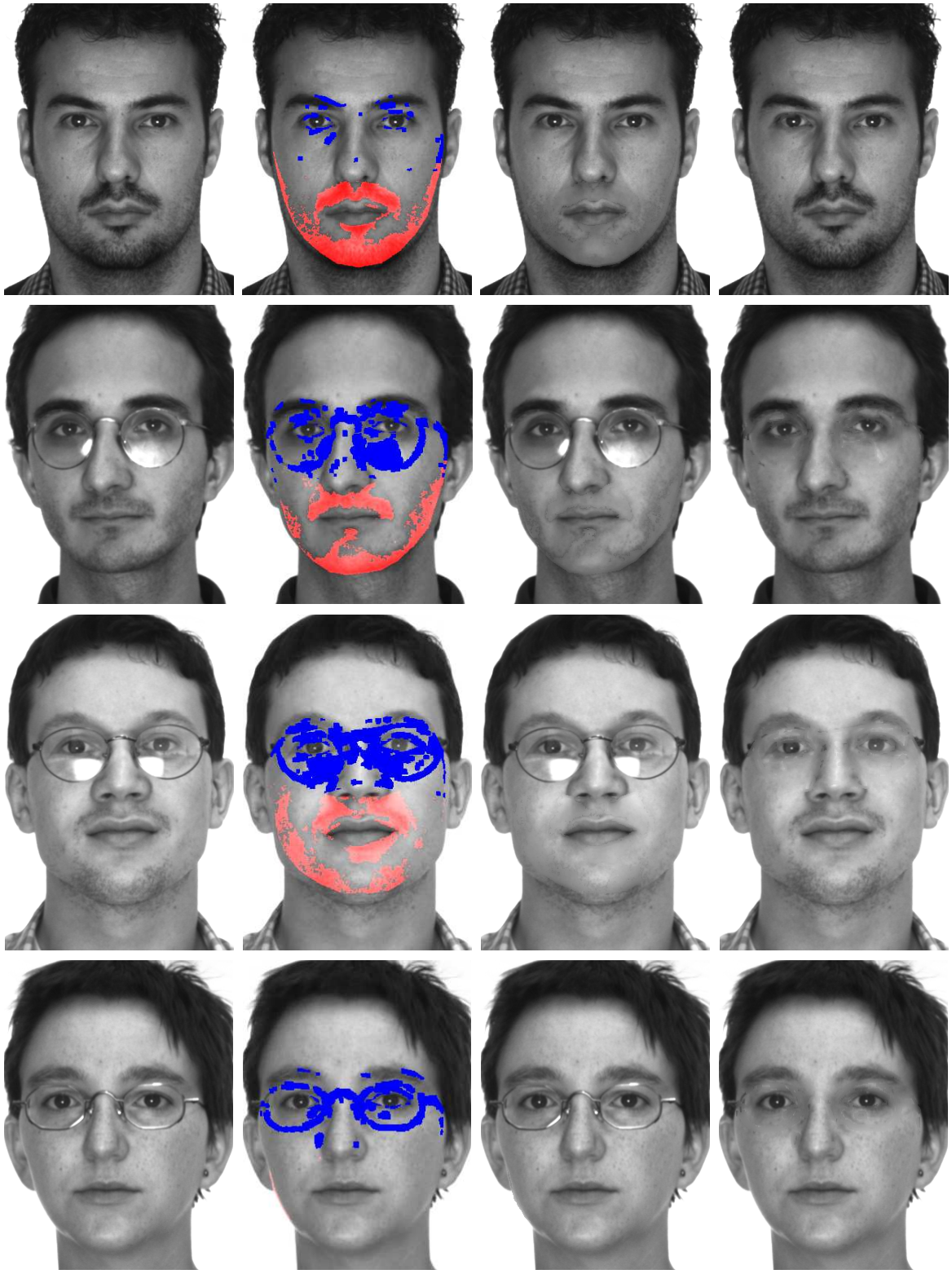


Figure 7.7: (a) originals (b) segmentation results (c) *free-from-hair* images (d) *free-from-glasses* images

Conclusion

The Active Appearance Models (AAMs) were proposed by Cootes *et al.* in 1998 and further improved in 2004 by Matthews *et al.* These models combine two statistical models, one to express shape deformations and the other to express appearance variations. Both are built from a training set of registered images that represent various instances of the object to model. AAMs have largely been used to model faces and to operate facial analysis on images and videos: an AAM can be automatically aligned onto a face and its features on an input picture. This alignment process is called *fitting*.

AAMs can provide impressive results in certain favorable contexts and the fitting can be of excellent accuracy, but this is not the case in all contexts, and especially when the face is unseen from the AAM, *i.e.* the face has not explicitely learnt by the AAM.

The AAMs are generally used without extended knowledge of there capabilities. In this thesis work we bring several contributions:

1. a better and clear understanding on the way an AAM can potentially perform in various contexts under its basic state-of-the-art formulation,
2. a better understanding on the way to best use an AAM in the context of unseen face fitting: the **specialization** of the AAM is a key to unseen face fitting,
3. a statistical-based measurement method allowing a more objective judgement of the fitting accuracy,
4. a method consisting to segment the AAM into sub-face region modelling AAMs to obtain what we call **local** or **segmented** AAMs, which further improve the fitting accuracy on unseen faces,
5. a promising investigation direction to face the highly complex problem of **pose and expression retrieval on unseen faces** and based on the use of a **cluster of segmented and specialized AAMs**,
6. an enhanced AAM formulation making it robust to illumination variation which is very common in many application contexts, but where the classical AAM is particularly unrobust. We call **Light-Invariant AAM** this AAM under the new formulation,
7. a useful method to **segment facial artefacts** (here glasses and beard as an example of use), thus giving them a **semantical meaning**, and to substitute these visual elements by their equivalent non-artefact pixels.

It is interesting to resume the work presented throughout this thesis.

In chapter 3 we give an overview of the various contexts of use of the method. An AAM performs best when the face to be fitted has been learnt by the model. This context is called *person-specific*. In this context we published [86], a first work where we extract facial expressions from a face explicitely learnt by the AAM.

In the same chapter we proposed to deeper study this context, and we analysed the performances of an AAM trained to fit a particular face, in a particular sequence. The inner settings of the AAM, as well as the quantity and nature of data it learns showed to have an influence on the fitting per-

formances. As a conclusion of this chapter, we observed that for an appropriate setting of the AAM, the model better performs when it is specialized on the fewest amount of data it has to fit onto. The more has to be learnt, the smaller are the performances.

This gave us an idea on how to maximize the performances of an AAM in the difficult context of unseen faces. The model has to be *specialized* as much as possible. A way to better specialize an AAM in the unseen context is to fix the head pose and the facial expression in order that only the identity variation has to be learnt. In chapter 4 we experimented this case and observed the fitting accuracy on unseen face presenting a frontal pose and a neutral expression, when the same kind of faces with different identity were learnt by the AAM. To measure the fitting accuracy we introduced the *Statistical Shape Error* that builds a statistics around multiple manual labelling on each face and allows more objective evaluation of the fitting accuracy of automatical and/or manual labellings. We first presented this measurement method in [77].

We observed a lack of fitting accuracy on unseen faces, and we highlighted the reason for it: the limited expressivity of the AAM appearance component is responsible for this lack.

Despite the specialization of the model, the fitting on unseen faces is then limited. To overcome this problem and increase the appearance expressivity of the model, we relied on smaller AAMs that we called segmented or local AAMs, in opposition with the whole face covering model that we called global AAM. We showed that smaller models present a higher generability power and are prone to provide a better fitting accuracy.

Unfortunately the local models are less robust than global AAMs and should be very well intialized on the input picture. To address this problem we presented a coarse-to-fine strategy to robustly fit the facial features with accuracy. First a global model is fitted onto the face, giving a good initialization for smaller models that we called intermediary models, each one modelling two features of the face. In turn, these models are fitted onto the picture and give a finer initialization for the local models, each one representing a single facial feature. The final fitting accuracy resulted very accurate on most of the tested faces, and it often reached a statistical error score comparable to manual labelling's. The main points of this chapter were presented in [87].

The positive results obtained with the use of specialized and local models led us to further extend the way AAMs can be used. In chapter 5 we conducted a preliminary test on unseen faces presenting varying pose and expression. We relied on a set, or cluster of AAMs trained over many identities, but specialized only for one pose and one facial deformation. Moreover each model focusses on the upper or on the lower part of the face, then having a higher generability than global models.

For each picture presenting an unseen face to test we ran all available AAMs. A simple least average absolute reconstruction error criteria is used to select the winner AAM for this test image. For as few as 8 training images, 80% trials showed that the winner AAM fitted the face with accuracy, and for 62% of the trials the winner AAM pose and expression it is specialized for was matching the current pose and deformation of the test face. A higher percentage could be obtained for accuracy if more identities were included in the training set. However, to obtain a significantly better result for the correct match between the pose and deformation of the winner AAM and the pose and deformation of the test face, the training database should be built more thoroughly. Further investigations in this direction will be done to understand its real limitation, and decide how to improve it further. We can maybe robustify the method with more robust selection criteria of the correct AAM, or even assist it with the decision of a bunch of machine learning based classifiers. Existing correlations between upper and lower facial deformations occuring during spontaneous expressions can also be included into a decision process aiming to detect some incoherent results.

In chapter 6 we successfully propose a solution to fit an AAM under varying illumination. The

performance of our solution were tested in the person-specific context. An interesting perspective would be to investigate the reliability of this solution in the case of unseen faces, thus extending the work done in chapters 4 and 5 to the case of light changes.

Furthermore, a collaboration with Hugo Mercier conducted to the following publication [78] where we overcome the state of the art solution of face fitting with AAM under occlusion proposed by Theobald *et al.* in [101]. Our study was done in the person-specific context, and and extension to unseen faces would also be the object of an interesting investigation.

Finally in chapter 7 we present a solution to segment facial artifacts on faces, like facial hair or glasses, and then to substitute the segmented pixels with their non-artefact equivalent. The results are convincing and useful for facial characterization, and the subtitution should be extended to color images to broaden the range of applications.

Annexe

A weighted sum of square residual value equation can generaly be written as follows:

$$\sum_x \boldsymbol{Q}(x).[\boldsymbol{g}(x) - \boldsymbol{G}^x \boldsymbol{p}]^2 \tag{4}$$

where $\boldsymbol{G}^x$ is the xth line of matrix $\boldsymbol{G}$ containing c row vectors that we linearly combine by assigning them the coefficients contained in the vector $\boldsymbol{p}$ to be estimated. The residual error commited on each element of the difference are weighted by the coefficients contained in the vector $\boldsymbol{Q}$.

To minimize the equation (4) it is useful to express it under a purely matricial form, so allowing to use the classical matricial derivation rules to find the solution for the optimum $\boldsymbol{p}$ vector.

The equation (4) becomes:

$$(\boldsymbol{g} - \boldsymbol{G}\boldsymbol{p})^T \boldsymbol{q}^T \boldsymbol{q}(\boldsymbol{g} - \boldsymbol{G}\boldsymbol{p}) \tag{5}$$

where $\boldsymbol{q}$ is the matrix

$$\boldsymbol{q} = diag(\boldsymbol{Q})^{\frac{1}{2}} = \begin{pmatrix} \sqrt{\boldsymbol{Q}(1)} & 0 & \cdots & 0 \\ 0 & \sqrt{\boldsymbol{Q}(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sqrt{\boldsymbol{Q}(N)} \end{pmatrix}$$

The equation (5) can be developed as follows:

$$(\boldsymbol{g}^T - \boldsymbol{p}^T \boldsymbol{G}^T)\boldsymbol{q}^T \boldsymbol{q}(\boldsymbol{g} - \boldsymbol{G}\boldsymbol{p})$$

$$= \boldsymbol{g}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{g} - \boldsymbol{g}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G}\boldsymbol{p} -$$

$$\boldsymbol{p}^T \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{g} - \boldsymbol{p}^T \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G}\boldsymbol{p}$$

Derivating this formula with respect to $\boldsymbol{p}$ gives

$$-\boldsymbol{g}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G} - \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{g} + 2\boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G}\boldsymbol{p} \tag{6}$$

Finaly looking for the null solution, we obtain the following closed-formed

$$\boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G}\boldsymbol{p} = \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{g} \tag{7}$$

If

$$\boldsymbol{A} = \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{G}$$

we have

$$\boldsymbol{p} = (\boldsymbol{A}^T \boldsymbol{A})^{-1} \boldsymbol{A}^T \boldsymbol{G}^T \boldsymbol{q}^T \boldsymbol{q}\boldsymbol{g} \tag{8}$$

Bibliography

[1] J. Ahlberg. Candide-3 - un updated parameterised face. Technical report, Dept. of Electrical Engineering, Linköping University, Sweden, January 2001.

[2] J. Ahlberg. An active model for facial feature tracking. In *Int. Workshop on Image Analysis for Multimedia Interactive Services*, Tampere, Finland, May 2001.

[3] A.Mehrabian. Communication without words. *Psychology Today*, 1968.

[4] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, November 2003.

[5] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 4. Technical Report CMU-RI-TR-04-14, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2004.

[6] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-Kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-35, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, February 2003.

[7] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. *Face Recognition Across Pose and Illumination*. Springer-Verlag, 2004.

[8] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1:1090–1097, 2001.

[9] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221 – 255, March 2004.

[10] S. Baker, I. Matthews, and J. Schneider. Automatic construction of active appearance models as an image coding problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1380–1384, October 2004.

[11] S. Baker, R. Patil, G. Cheung, and I. Matthews. Lucas-Kanade 20 years on: Part 5. Technical Report CMU-RI-TR-04-64, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2005.

[12] M.S. Bartlett, G.L. Donato, J.R. Movellan, J.C. Hager, P. Ekman, and T.J. Sejnowski. Image representations for facial expression coding. *Advances in Neural Information Processing Systems*, 12:886–892, 2000.

[13] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscek, I. Fasel, M.G. Frank, and J. Movellan. Automatic recognition of facial actions in spontaneous expressions. *Journal of Multimedia*, 2006.

[14] M.S. Bartlett, G.C. Littlewort, B. Braathen, E. Smith, and J.R. Movellan. An approach to automatic analysis of spontaneous facial expressions. In *Proceedings of the 5th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'02)*, page 360, Washington, D.C, 2002.

[15] M.S. Bartlett, G.C. Littlewort, C. Lainscek, I. Fasel, M.G. Frank, and J.R. Movellan. Fully automatic facial action recognition in spontaneous behavior. *7th International Conference on Automatic Face and Gesture Recognition*, pages 223–228, 2006.

[16] M.S. Bartlett, G.C. Littlewort, C. Lainscek, I. Fasel, and J.R. Movellan. Machine learning methods for fully automatic recognition of facial expressions and facial actions. *Proceedings of the IEEE Conference on Systems, man and Cybernetics, The Hague, Netherlands*, 2004.

[17] M.S. Bartlett, J.R. Movellan, G.C. Littlewort, B. Braathen, M.G. Frank, and T.J. Sejnowski. Towards automatic recognition of spontaneous facial actions, 2005.

[18] A. Bartoli. Groupwise geometric and photometric direct image registration. In *BMVC'06 - Proceedings of the Seventeenth British Machine Vision Conference, Edinburgh, UK*, September 2006.

[19] M.J. Black and Y. Yacoob. Tracking and recognizing facial expressions in image sequences, using local parameterized models of image motion. Technical Report CS-TR-3401, University of Maryland, January 1995.

[20] M.J. Black and Y. Yacoob. Recognizing facial expression in image sequences using local parameterized models of image motion. *Trans. Computer Vision*, 1:23–48, 1997.

[21] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D-faces. *SIGGRAPH 99*, 1999.

[22] J. M. Buenaposada. *Análisis de expresiones faciales mediante visión por computador*. PhD thesis, Universidad Politécnica de Madrid, 2005.

[23] J. Cohn, A. Zlochower, J. Lien, and T. Kanade. Feature-point tracking by optical flow discriminates subtle differences in facial expression, 1998.

[24] Jeffrey F. Cohn and Takeo Kanade. Automated facial image analysis for measurement of emotion expression, 2004.

[25] Jeffrey F. Cohn, A. J. Zlochower, J. Lien, and Takeo Kanade. Automated face analysis by feature point tracking has high concurrent validity with manual faces coding. *Psychophysiology*, 36:35–43, 1999.

[26] T. Cootes, D. H. Cooper, C.J. Taylor, and J. Graham. A trainable method of parametric shape description. *Proc. British Machine Vision Conference*, 1991.

[27] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. *In Proceedings of the European Conference on Computer Vision*, 2:484–498, 1998.

[28] T. Cootes, G. Edwards, and C.J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, june 2001.

[29] T. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - smart snakes. *Proc. British Machine Vision Conference*, 1992.

[30] T. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. *Proc. British Machine Vision Conference*, 1992.

[31] T. Cootes, C. J. Taylor, D.H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision Graphics and Image Understanding*, 61(1):38–59, 1995.

[32] T. Cootes, C. J. Taylor, and A. Lanitis. Multi-resolution search using active shape models. *Proc. 12th International Conference of Pattern Recognition*, 1:610–612, 1994.

[33] T. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham. Building and using flexible models incorporating grey-level information. *Proc. of the 4th ICCV*, pages 355–365, 1993.

[34] T. Cootes and C.J. Taylor. Statistical models of appearance for computer vision, 1999.

[35] K.D. Craig, S.A. Hyde, and C.J. Patrick. Genuine, suppressed, and faked facial behavior during exacerbation of chronic low back pain. *Pain*, pages 46:161–172, 1991.

[36] I. Craw and P. Cameron. Parameterizing images for recognition and reconstruction. *In Proc. BMVC*, 1991.

[37] C. Darwin. *The expression of emotions in man and animals.* Oxford University Press, New York, 1872/1998.

[38] R. Descartes. *Les passions de l'âme.* Henry Le Gras, 1649.

[39] I. L. Dryden and K. V. Mardia. *Statistical shape analysis.* John Wiley & Sons, New York, 1998.

[40] P. Ekman. Facial expression of emotion. *American Psychologist*, pages 48:384–392, 1993.

[41] P. Ekman, R.J. Davidson, and W.V. Friesen. The duchenne smile: Emotional expression and brain physiology. *Journal of Personality and Social Psychology*, pages 58:342–353, 1990.

[42] P. Ekman and W. V. Friesen. *Facial Action Coding System (FACS) : Manual.* Palo Alto: Consulting Psychologists Press, 1978.

[43] P. Ekman and W.V. Friesen. Felt, false, and miserable smiles. *Journal of Nonverbal Behavior*, pages 6:238–252, 1982.

[44] P. Ekman, W.V. Friesen, and P. Ellsworth. Emotion in the human face. *Pergamon Press, New York*, 1972.

[45] P. Ekman and E.L. Rosenberg. *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression using the Facial Action Coding System (FACS).* Oxford University Press, New York, 1997.

[46] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):757–763, 1997.

[47] B. Fasel and J. Luettin. Automatic facial expression analysis : A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[48] I. Fasel, B. Fortenberry, and J. R. Movellan. Generative framework for real-time object detection and classification. *Computer Vision and Image Understanding*, 98(1):182–210, 2004.

[49] G.D. Finlayson, M.S.Drew, and C.Lu. Intrisic images by entropy minimization. In *ECCV'04 - European Conference on Computer Vision*, 2004.

[50] G.D. Finlayson, S.D.Hordley, and M.S.Drew. Removing shadows from images. In *ECCV'02 - European Conference on Computer Vision*, 2004.

[51] M.G. Franck and P. Ekman. Appearing truthful generalized across different deception situations. *Journal of Personality and Social Psychology*, pages 86:486–495, 2004.

[52] M.G. Frank. International encyclopedia of the social and behavioral sciences. *chapter Facial expressions. Elsvier, Oxford*, 2002.

[53] M.G. Frank and P. Ekman. Not all smiles are created equal: The differences between enjoyment and other smiles. *Humor: The International Journal for Research in Humor*, pages 6:9–26, 1993.

[54] V. Gay-Bellile, A. Bartoli, and P. Sayd. Feature-driven non-rigid image registration. In *BMVC'07 - Proceedings of the Eighteenth British Machine Vision Conference, Warwick, UK.*, September 2007.

[55] J. Gonzalez, F. De la Torre Frade, R. Murthi, N. Guil Mata, and E. Zapata. Bilinear active appearance models. In *ICCV'07 - Workshop on Non-rigid Registration and Tracking through Learning*, October 2007.

[56] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *British Machine Vision Conference*, 2004.

[57] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(11):1080–1093, November 2005.

[58] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025 – 1039, 1998.

[59] S. Hamlaoui and F. Davoine. Facial action tracking using an aam-based condensation approach. *IEEE International Conference on Acoustics, Speech, and Signal Processing, Philadelphia, PA, USA*, 2005.

[60] Z. Hammal. *Segmentation des Traits du Visage, Analyse et Reconnaissance d'Expressions Faciales par les Modëles de Croyance Transférable.* PhD thesis, Université Joseph Fourier Grenoble, june 2006.

[61] M. Heller and V. Haynal. Les visages de la dépression de suicide. *Cahiers Psychiatrique Genevois (Medecin et Hygiene Editors)*, pages 16:107–117, 1994.

[62] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 1998.

[63] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990.

[64] S.C. Koterba, S. Baker, I. Matthews, C. Hu, J. Xiao, J. Cohn, and T. Kanade. Multi-view aam fitting and camera calibration. In *ICCV'05 - Proc. International Conference on Computer Vision*, October 2005.

[65] F. De la Torre. Automatic learning of appearenace face models. *Second International Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-time Systems, Vancouver*, 2001.

[66] F. De la Torre Frade, A. Collet Romea, J. Cohn, and T. Kanade. Filtered component analysis to increase robustness to local minima in appearance models. In *CVPR'07 - IEEE Conference on Computer Vision and Pattern Recognition*, June 2007.

[67] A. Lanitis, C. J. Taylor, and T. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, July 1997.

[68] R. Lanzarotti. *Facial feature detection and description.* PhD thesis, Universitá Degli Studi Di Milano, 2003.

[69] J.J. Lien, T. Kanade, J.F. Cohn, and C. Li. Subtly different facial expression recognition and expression intensity estimation. *Proc. Computer Vision and Pattern Recognition (CVPR)*, pages 853–859, June 23-25 Santa Barbara, CA 1998.

[70] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Procedings of International Joint Conference on Artificial Intelligence, Vancouver, Canada*, 18(11):674–680, August 1981.

[71] S. Lucey, I. Matthews, C. Hu, Z. Ambadar, F De la Torre Frade, and J. Cohn. Aam derived face representation for robust facial action recognition. *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 155–160, 2006.

[72] S. Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, 1999.

[73] A.M. Martinez and R. Benavente. The AR face database. Technical Report 24, CVC, June 1998.

[74] K. Mase and A. Pentland. Lipreading by optical flow. systems and computers. *Systems and Computers*, pages 22(6):67–76, 1991.

[75] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, November 2004.

[76] H. Mercier. *Modelisation et suivi des deformations faciales.* PhD thesis, IRIT - Toulouse, 2007.

[77] H. Mercier, J. Peyras, and P. Dalle. Toward an efficient and accurate AAM fitting on appearance varying faces. *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 363–368, 2006.

[78] H. Mercier, J. Peyras, and P. Dalle. Occluded facial expression tracking. In *SCIA'07, Scandinavian Conference on Image Analysis, Aalborg, Denmark.*, June 2007.

[79] T. Moriyama, J. Xiao, J.F. Cohn, and T. Kanade. Meticulously detailed eye model and its application to analysis of facial image. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, pages 629 – 634, 2004.

[80] E. Mu noz, J. M. Buenaposada, and L. Baumela. Efficient model-based 3d tracking of deformable objects. *International Conference on Computer Vision*, 1:877–882, 2005.

[81] 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 - Part B*, 36(2):433–449, April 2006.

[82] M. Pantic and L. J. M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[83] M. Pantic and L. J. M. Rothkrantz. Expert system for automatic analysis of facial expressions. *Image and Vision Computing Journal*, 18(11):881–905, 2000.

[84] M. Pantic and L. J. M. Rothkrantz. Towards an affect-sensitive multimodal human-computer interaction. *in Proceedings of the IEEE*, 91(9):1370–1390, 2003.

[85] J-S. Park, S.C. Ahn Y.H. Oh, and S.W. Lee. Glasses removal from facial image using recursive error compensation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(5):805–811, may 2005.

[86] J. Peyras. Extraction des expressions faciales en présence de rotations hors-plan. *17e Conference Francophone sur l'Interaction Homme-Machine*, 2005.

[87] J. Peyras, A. Bartoli, H. Mercier, and P. Dalle. Segmented aams improve person-independent face fitting. In *BMVC'07 - Proceedings of the Eighteenth British Machine Vision Conference, Warwick, UK.*, September 2007.

[88] J. Peyras, H. Mercier, and P. Dalle. Segmentation des artéfacts du visage par utilisation de modèles à apparence active. *ORASIS'07*, june 2007.

[89] R.W. Picard. *Affective Computing*. MIT Press, 1997.

[90] D. Pizarro and A. Bartoli. Shadow resistant direct image registration. In *SCIA'07 - Proceedings of the Fifteenth Scandinavian Conference on Image Analysis, Aalborg, Denmark*, June 2007.

[91] S. Romdhani. *Face Image Analysis using a Multiple Features Fitting Strategy*. PhD thesis, Universität Basel, January 2005. Inauguraldissertation.

[92] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International conference on Computer Vision 2003*, 2003.

[93] E.L. Rosenberg, P. Ekman, W. Jiang, and M. Babyak. Linkages between facial expressions of anger and transient myocardial ischemia in men with coronary artery disease. *American Psychological Assn, US.Emotion*, pages 1(2):107–115, 2001.

[94] M. Rosenblum, Y. Yacoob, and L.S. Davis. Human expression recognition from motion using a radial basisfunction network architecture. *IEEE Transactions on Neural Networks*, 7:1121–1138, Sep 1996.

[95] H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, page 38, Santa Barbara, CA, June 1998.

[96] M.A. Sayette, D.W. Smith, M.J. Breiner, and G.T. Wilson. The effect of alcohol on emotional response to a social stressor. *Journal of Studies on Alcohol*, pages 53:541–545, 1992.

[97] S. Sclaroff and J. Isidoro. Active blobs. *In Proceedings of the 6th ICCV*, 1996.

[98] H. Y. Shum and R. Szeliski. Construction of a panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 16(1):63–84, 2000.

[99] G. Silveira and E. Malis. Real-time visual tracking under arbitrary illumination changes. In *CVPR'07 - IEEE Computer Vision and Pattern Recognition. Minneapolis, USA*, June 2007.

[100] M. Tekalp. Face and 2d mesh animation in mpeg-4. *Tutorial Issue on the MPEG-4 Standard, Image Communication Journal, Elsevier*, 1999.

[101] B.J. Theobald, I. Matthews, and S. Baker. Evaluating error functions for robust active appearance models. *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, page 149154, 2006.

[102] Y. Tian, T. Kanade, and J.F. Cohn. Evaluation of gabor-wavelet-based facial action unit recognition inimage sequences of increasing complexity. *Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 218–223, may 2002.

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

[104] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2007.

[105] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[106] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2002.

[107] J. Whitehill and C. W. Omlin. Haar features for facs au recognition. *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 357–362, 2006.

[108] J. Whitehill and C. W. Omlin. Local versus global segmentation for facial expression recognition. *Proc. of the 7th IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 97–101, 2006.

[109] C. Wu, C. Liu, H-Y. Shum, Y-Q. Xu, and Z. Zhang. Automatic eyeglasses removal from face images. *The 5th Asian Conference on Computer Vision*, 26(3):322–336, january 2002.

[110] J. Xiao, S. Baker, I. Matthews, and T. Kanade. Real-time combined 2d+3d active appearance models. In *CVPR'07 - IEEE Computer Vision and Pattern Recognition. Washington, DC, USA*, June 2004.

[111] J. Xiao, T. Kanade, and J. F. Cohn. Robust full-motion recovery of head by dynamic. *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pages 163–169, 2002.

[112] X.Liu, F.W.Wheeler, and P.H.Tu. Improved face model fitting on video sequences. In *BMVC'07 - Proceedings of the Seventeenth British Machine Vision Conference, Warwick, UK*, September 2007.

[113] X.Liu, P.H.Tu, and F.W.Wheeler. Face model fitting on low resolution images. In *BMVC'06 - Proceedings of the Seventeenth British Machine Vision Conference, Warwick, UK*, September 2006.

[114] Y. Yacoob and L.S. Davis. Computing spatio-temporal representations of human faces. In *CVPR94*, pages 70–75, 1994.

[115] Y. Yacoob and L.S. Davis. Recognizing human facial expressions from long image sequences using optical flow. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, pages 6:636–642, June 1996.

[116] Z. Zhang. Feature-based facial expression recognition: Experiments with a multi-layer perceptron. Technical Report 3354, Institut National de Recherche en Informatique et en Automatique, Février 1998.

[117] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu. Comparison between geometry-based and gaborwavelets-based facial expression recognition using multi-layer perceptron. *in Proc. Third IEEE International Conference on Automatic Face and Gesture Recognition (FG'98)*, April 1998.

[118] Y. Zhou, L. Gu, and H.J. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. *CVPR*, 2003.