Department of Electrical and Computer Engineering
Faculty of Sciences and Technology
University of Coimbra

Parametric Face Alignment:
Generative and Discriminative Approaches

Pedro Alexandre Dias Martins
http://www.isr.uc.pt/~pedromartins
pedromartins@isr.uc.pt

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

September 2012

Advisor: Professor Jorge Batista

Abstract

This thesis addresses the matching of deformable human face models into 2D images. Two different approaches are detailed: generative and discriminative methods. Generative or holistic methods model the appearance/texture of all image pixels describing the face by synthesizing the expected appearance (it builds synthetic versions of the target face). Discriminative or patch-based methods model the local correlations between pixel values. Such approach uses an ensemble of local feature detectors all connected by a shape regularization model. Typically, generative approaches can achieve higher fitting accuracy, but discriminative methods perform a lot better in unseen images.

The Active Appearance Models (AAMs) are probably the most widely used generative technique. AAMs match parametric models of shape and appearance into new images by solving a nonlinear optimization that minimizes the difference between a synthetic template and the real appearance. The first part of this thesis describes the 2.5D AAM, an extension of the original 2D AAM that deals with a full perspective projection model. The 2.5D AAM uses a 3D Point Distribution Model (PDM) and a 2D appearance model whose control points are defined by a perspective projection of the PDM. Two model fitting algorithms and their computational efficient approximations are proposed: the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA). Robust solutions for the SFA and NFA are also proposed in order to take into account the self-occlusion and/or partial occlusion of the face. Extensive results, involving the fitting convergence, fitting performance in unseen data, robustness to occlusion, tracking performance and pose estimation are shown.

The second main part of this thesis concerns to discriminative methods such as the Constrained Local Models (CLM) or the Active Shape Models (ASM), where an ensemble of local feature detectors are constrained to lie within the subspace spanned by a PDM. Fitting such a model to an image typically involves two steps: (1) a local search using a detector, obtaining response maps for each landmark and (2) a global optimization that finds the shape parameters that jointly maximize all the detection responses. This work proposes: Discriminative Bayesian Active Shape Models (DBASM) a new global optimization strategy, using a Bayesian approach, where the posterior distribution of the shape parameters are inferred in a maximum a posteriori (MAP) sense by means of a Linear Dynamical System (LDS). The DBASM approach models the covariance of the latent variables i.e. it uses 2^{nd} order statistics of the shape (and pose) parameters. Later, Bayesian Active Shape Models (BASM) is presented. BASM is an extension of the previous DBASM formulation where the prior distribution is explicitly modeled by means of recursive Bayesian estimation. Extensive results are presented, evaluating DBASM and BASM global optimization strategies, local face parts detectors and tracking performance in several standard datasets. Qualitative results taken from the challenging Labeled Faces in the Wild (LFW) dataset are also shown.

Finally, the last part of this thesis, addresses the identity and facial expression recognition. Face geometry is extracted from input images using the AAM and low dimensional manifolds were then derived using Laplacian EigenMaps (LE) resulting in two types of manifolds, one for representing identity and the other for person-specific facial expression. The identity and facial expression recognition system uses a two stage approach: First, a Support Vector Machines (SVM) is used to establish identity across expression changes, then the second stage deals with person-specific expression recognition with a network of Hidden Markov Models (HMMs). Results taken from people exhibiting the six basic expressions (happiness, sadness, anger, fear, surprise and disgust) plus the neutral emotion are shown.

Keywords:

Generative Methods; Discriminative Methods; Non-Rigid Face Registration; Image Alignment; Active Appearance Models (AAM); Active Shape Models (ASM); Identity Recognition; Facial Expression Recognition.

Resumo

Esta tese aborda a correspondência de modelos humanos de faces deformáveis em imagens 2D. São apresentadas duas abordagens diferentes: métodos generativos e discriminativos. Os modelos generativos ou holísticos modelam a aparência/textura de todos os pixeis que descrevem a face, sintetizando a aparência esperada (são criadas versões sintéticas da face alvo). Os modelos discriminativos ou baseados em partes modelam correlações locais entre valores de pixeis. Esta abordagem utiliza um conjunto de detectores locais de características, conectados por um modelo de regularização geométrico. Normalmente, as abordagens generativas permitem obter uma maior precisão de ajuste do modelo, mas os métodos discriminativos funcionam bastante melhor em imagens nunca antes vistas.

Os Modelos Activos de Aparência (AAMs) são provavelmente a técnica generativa mais utilizada. Os AAMs ajustam modelos paramétricos de forma e aparência em imagens, resolvendo uma optimização não linear que minimiza a diferença entre o modelo sintético e a aparência real. A primeira parte desta tese descreve os AAM 2.5D, uma extensão do AAM original 2D que permite a utilização de um modelo de projecção em perspectiva. Os AAM 2.5D utilizam um Modelo de Distribuição de Pointos (PDM) e um modelo de aparência 2D cujos pontos de controlo são definidos por uma projecção em perspectiva do PDM. Dois algoritmos de ajuste do modelo e as suas aproximações eficientes são propostas: *Simultaneous Forwards Additive (SFA)* e o *Normalization Forwards Additive (NFA)*. Soluções robustas para o SFA e NFA, que contemplam a oclusão parcial da face, são igualmente propostas. Resultados extensos, envolvendo a convergência de ajuste, o desempenho em imagens nunca vistas, robustez à oclusão, desempenho de seguimento e estimativa de pose são apresentados.

A segunda parte desta da tese diz respeito os métodos discriminativos, tais como os Modelos Locais com Restrições (CLM) ou os Modelos Activos de Forma (ASM), onde um conjunto de detectores de caracteristicas locais estão restritos a pertencer ao subespaço gerado por um PDM. O ajuste de um modelo deste tipo, envolve tipicamente duas etápas: (1) uma pesquisa local utilizando um detector, obtendo mapas de resposta para cada ponto de referência e (2) uma estratégia de optimização global que encontra os parâmetros do PDM que permitem maximizar todas as respostas conjuntamente. Neste trabalho é proposto o *Discriminative Bayesian Active Shape Models (DBASM)*, uma nova estratégia de optimização global que utiliza uma abordagem Bayesiana, onde a distribuição *a posteriori* dos parâmetros de forma são inferidos por meio de um sistema dinâmico linear. A abordagem DBASM modela a covariância das variáveis latentes ou seja, é utilizado estatística de segunda ordem na modelação dos parâmetros. Posteriormente é apresentada a formulação *Bayesian Active Shape Models (BASM)*. O BASM é uma extensão do DBASM, onde a distribuição *a priori* é explicitamente modelada por meio de estimação Bayesiana recursiva. São apresentados resultados extensos, avaliando as estratégias de optimização globais DBASM e BASM, detectores locais de componentes da face, e desempenho de seguimento em várias bases de dados padrão. Resultados qualitativos extraídos da desafiante base de dados *Labeled Faces in the Wild (LFW)* são também apresentados.

Finalmente, a última parte desta tese aborda o reconhecimento de idêntidade e expressões faciais. A geometria da face é extraída de imagens utilizando o AAM e variedades de baixa dimensionalidade são derivadas utilizando *Laplacian EigenMaps (LE)*, obtendo-se dois tipos de variedades, uma para representar a idêntidade e a outra para expressões faciais específicas de cada pessoa. A idêntidade e o sistema de reconhecimento de expressões faciais utiliza uma abordagem de duas fases: Num primeiro estágio é utilizado uma Máquina de Vectores de Suporte (SVM) para determinar a idêntidade, dedicando-se o segundo estágio ao reconhecimento de expressões. Este estágio é especifico para cada pessoa e utiliza Modelos de Markov Escondidos (HMM). São mostrados resultados obtidos em pessoas exibindo as seis expressões básicas (alegria, tristeza, raiva, medo, surpresa e nojo), e ainda a emoção neutra.

Acknowledgements

In first, I would like to thank Professor Jorge Batista for accepting me as his PhD student, for his support and supervision. I also wish to express acknowledgements to all my laboratory colleagues, in special to Rui Caseiro and João Henriques for their friendly support, helpful ideias and useful comments.

This thesis was carried out at the Institute of Systems and Robotics (ISR) in the Department of Electrical and Computer Engineering (DEEC) from the Faculty of Sciences and Technology of the University of Coimbra (http://www.isr.uc.pt).

The research in this thesis was supported by the Portuguese Science Foundation, *Fundação para a Ciência e a Tecnologia (FCT)*, with the PhD scollarship grant SFRH/BD/45178/2008. Additionally, I also acknowledge the FCT for its support to our research group through the project: Identity Recognition using 4D-Facial Dynamics with grant PTDC/EIA-CCO/108791/2008 (http://dynface4d.isr.uc.pt).

This section should also mention Joana (that's it!). Finally, I would like to thank to my parents for their long lasting support.

FCT
Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

Contents

List of Figures

List of Tables

List of Abbreviations

AAM	–	Active Appearance Models
ASM	–	Active Shape Models
BASM	–	Bayesian Active Shape Models
CLM	–	Constrained Local Models
DOF	–	Degrees of Freedom
DBASM	–	Discriminative Bayesian Active Shape Models
ENFA	–	Efficient Normalization Forwards Additive
ESFA	–	Efficient Simultaneous Forwards Additive
ERNFA	–	Efficient Robust Normalization Forwards Additive
ERSFA	–	Efficient Robust Simultaneous Forwards Additive
FFT	–	Fast Fourier Transform
GPA	–	Generalized Procrustes Analysis
HMM	–	Hidden Markov Models
LE	–	Laplacian EigenMaps
MAP	–	Maximum a Posteriori
NFA	–	Normalization Forwards Additive
PCA	–	Principal Component Analysis
PDM	–	Point Distribution Model
POSIT	–	Pose from Orthography and Scaling with ITerations
RNFA	–	Robust Normalization Forwards Additive
RSFA	–	Robust Simultaneous Forwards Additive
SFA	–	Simultaneous Forwards Additive
SIC	–	Simultaneous Inverse Compositional
SVM	–	Support Vector Machines

Chapter 1

Introduction

Non-rigid face registration or face alignment is the task of matching the shape of an unseen face with the shape of a canonical model (a reference template). Such a registration (i.e. correspondence between images) is the key aspect in many modern computer vision applications, such as advanced human computer interaction, face recognition, facial expression analysis, tracking, head pose, gaze estimation, image coding or realistic graphical animation. Facial registration is a very challenging task because images of faces present a large variability in shape, texture, pose and lighting conditions. Furthermore, there is a high degree of complexity needed by a computer system to understand what is a face. What does it mean? how does a face looks like? How does it change by expression and pose variation? How can the face structure be extracted? Nevertheless, the recovered information must be of some manageable size, because to process raw data is unfeasible or requires a lot of computational effort.

In this thesis, face models were used to entirely describe the face characteristics. Generally, a face model is a system where a set of input parameters generate a face output. Model based techniques represent a promising approach where a model representing the phenomenon of interest is matched with unseen data by setting its parameters accordingly. This kind of models can provide a high registration quality and simultaneously are able to represent the relevant facial information using a compact model (i.e. with only a small set of parameters) which is significant to further recognition tasks.

Both identity and facial expression recognition problems are also discussed later in this thesis. In typical applications, once the face model has been constructed, the first step is to fit it to an input image, i.e. to find the face model parameters that maximize the match between the model and the input image. Finally, the model parameters can be passed to additional stages where several purposes are possible (e.g. measurement, classification/recognition tasks).

1.1 Motivation

Aligning images of faces, using computer vision techniques, is a hard task. The human face can exhibit large amounts of variability, such as variations of identity, expression, pose, lighting, non-rigid motion and it can also present possibly incomplete evidence (i.e. occlusion, that could be partial or self-occlusion due to its 3D structure). Model-based methods offer possible solutions to these difficulties. Prior knowledge of a face can be used to understand the confusion caused by its structural complexity, adding extra tolerance to noisy or missing data. Since human faces are not all alike, we need do deal with variability, which leads to deformable models.

The Point Distribution Model (PDM) [97] generates shapes that maintain its main characteristics. It can deform to fit a range of examples and it can be constrained to generate only plausible shapes, all this using a simple linear parametrization. More important, given enough training data, the PDM is able to model unseen faces [79] (i.e. to behave like a generic face model). However, the same can not be said for the appearance/texture of a face. The variability in appearance, even for a small number of faces, can reach a huge dimensional representation (e.g. a 10 by 10 sized image can represent $2^{8\times10\times10}$ possible combination of grey values - the world population is less than 2^{33}).

In this context, two main appearance representation paradigms can be considered: the **generative** and the **discriminative** approaches. Briefly, the generative methods [99][38][100][105] are parametric models of appearance, similar to EigenFaces [59], that are able to synthesize new instances. In fact, the matching a generative model consists of building synthetic versions of the target face. The discriminative approaches

[98][24][23][115][41], on the other hand, use an ensemble of feature detectors that aim to search for each of the facial components. See a detailed explanation at section 1.2.

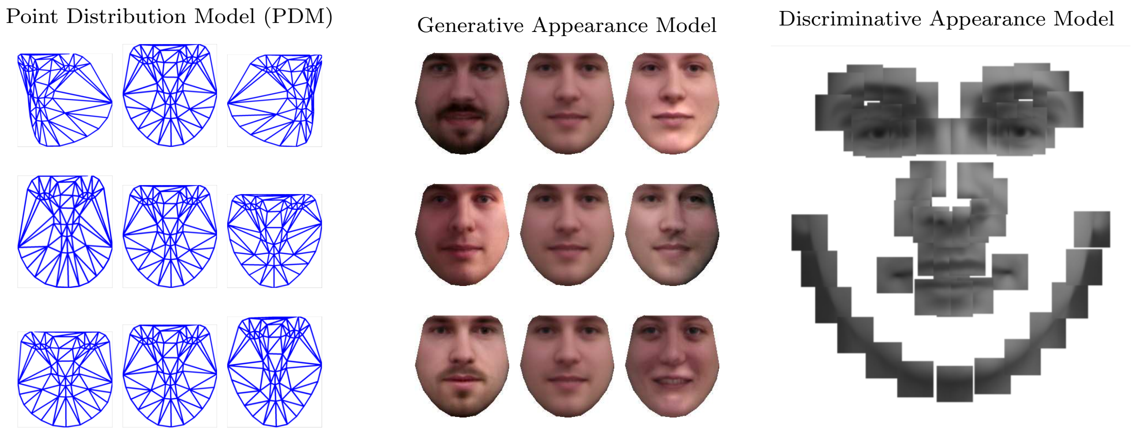
Building on the previous work [65], this thesis studies three important topics related to deformable face models: (1) enhancement of generative aligning methods; (2) new discriminative approach(es) and (3) identity and facial expression recognition tasks.

1.2 Generative vs Discriminative Approaches

The goal of parametric deformable fitting is to find the Point Distribution Model (PDM) [97] parameters that best describe a face in a target image. The PDM, or the shape model, is a linear parametric statistical technique that explains the geometric variation of shapes. The shape itself is defined as the quality of the configuration of points (landmarks) which are invariant under some transformation, usually a similarity transformation (i.e. translation, rotation and scaling) in the 2D or 3D cases. The landmarks are commonly selected to be structural important features of the face components (e.g. eye corners, eyebrows, mouth, nose, chin, etc) and the shape model is learnt from a representative dataset. It consists of a normalization step, using Generalized Procrustes Analysis (GPA) [15] that removes the similarity effects, followed by a Principal Components Analysis (PCA) [56]. Shape generation is based on a weighted linear combination of the PCA eigenvectors where the shape parameters represent these mixing weights.

Several model fitting strategies have been proposed, most of which can be categorized as being either generative (holistic) or discriminative (patch-based). These differences are highlighted in figure 1.1 where examples of PDM shapes, generative appearance and discriminative model are shown.

The holistic representations model the appearance of all image pixels describing the face. The Active Appearance Models (AAMs) [99][38] are probably the most widely used generative technique. AAMs match parametric models of shape (the PDM) and appearance (also captured by a PCA) into new images by solving a nonlinear optimization that minimizes the difference between the synthetic template and real appearance



Point Distribution Model (PDM)

Generative Appearance Model

Discriminative Appearance Model

Figure 1.1: Parametric face alignment aims to find the parameters of a Point Distribution Model (PDM) that best describe the face of interest in an image. The left image shows shape instances of the PDM (the first three modes of deformation). The appearance/texture model could be either generative or discriminative. In a generative representation all the pixels belonging to the face are modeled (center image). Discriminative methods only consider local correlation between pixels (right image).

(it builds synthetic versions of the target face). By synthesizing the expected appearance template it achieves a high registration accuracy on the dataset it was trained for but it performs poorly in unseen data (individuals not captured by the texture PCA). If the appearance of a target individual does not lie in the subspace spanned by the appearance basis, the AAM can not generate a 'good' template and the model fitting will not converge. The generative representation generalizes poorly when new faces of interest exhibits large amounts of variability such as variations of identity, expression, pose, lighting or non-rigid motion, due to the huge dimensional representation of the appearance (learnt from limited data). This generalization problem gets worse by the typically quadratic error norm (L2 norm) used. New data, that can be seen as outliers, have a significant impact on the fitting quality. In fact, some solutions [80][54] that deal with this generalization problem use robust error norms.

Discriminative or parts-based methods such as the Constrained Local Models (CLM) [23][25] or the Active Shape Models (ASM) [93][98], can improve the model's representation capacity, as it accounts only for local correlations between pixel values. Such approaches use an ensemble of local feature detectors (each landmark has it own expert detector), whose locations are constrained to lie within the subspace spanned by the shape model (PDM), i.e. independent landmark measurements are made which are then combined by enforcing a prior over their joint motion. Fitting such model involves a two step fitting strategy: a local search and a global optimization. The first step performs an exhaustive local search using a feature detector, obtaining response maps for each landmark. Then, the global optimization finds the PDM parameters that jointly maximize the detection responses at once.

In summary, generative approaches can achieve higher fitting accuracy, but discriminative methods perform a lot better in unseen images.

1.3 Related Work

Model-based deformable models that are able to fit to new data instances have great interest in computer vision. The contents of this thesis relates to techniques based on Point Distribution Models (PDM). Several approaches have been proposed.

Generative Methods: The Active Contours Models (ACMs) or Snakes [55] are energy minimizing models, that deform spline curves according to internal and external forces. Internal forces keep the curve smooth while the external forces pull the curve towards the local image features like lines or edges. In Active Blobs [91] the shape deformation is based on physical properties such as stiffness and elasticity modeled by Finite Element Methods (FEM). A single static texture template with illumination modeling is used. The Active Appearance Models (AAM) [99], described earlier in section 1.2, match parametric models of shape (PDM) and appearance (EigenFaces [59]), minimizing the texture difference between the model and the covered target by sets of image alignment warps (piecewise affine warps). The original fitting process rely on a precomputed regression matrix. Several AAM extensions exist. In Direct Appearance Models (DAM) [107] the authors noticed that one shape may contain many textures but no texture is associated with more than one shape. The DAM predicts the shape directly from texture information. In Constrained Active Appearance Models [94] the model matching is driven by a probabilistic framework (MAP formulation) allowing to include prior constraints on point positions. Adaptive AAMs [9][96] update the Jacobian matrix (considered fixed in the standard formulation [99]) improving fitting performances for individuals outside the training set. In [9] linearly adapts the gradient matrix during the convergence and [96] uses Quasi-Newton based methods.

The AAM has been reformulated with true analytical derived gradients by Matthews *et al.*[38], achieving a better fitting accuracy and real-time performances using the Inverse Compositional (IC) [83][84] approach. The Inverse Compositional method showed that the image registration, which involves solving a nonlinear optimization, can be done more efficiently using (inverse) compositional updates of the parameters (instead of the original additive updates [11]). By reversing the roles of the image and the model in the error function, both Jacobian and the Hessian matrices become constant and can be precomputed. Several AAM fitting algorithm, using the IC approach, have been proposed: the Simultaneous Inverse Compositional (SIC) [85], the Normalization Inverse Compositional (NIC) [84] and the Project-Out (PO) [38]. The main difference between them is the way the optimization is done, i.e. optimizing shape and appearance parameters at once (SIC) or to project out the appearance variation optimizing only the shape parameters (NIC, PO). The appearance variation effects can be removed from the error image (NIC) or from the steepest descent images (PO). In this last case, the optimization is done in a subspace in which the appearance variation can be ignored. Adaptive Active Appearance Model (AAAM) [111][110] introduce a modification in the cost function that includes aligning multiple (previously aligned) frames as an additional constrain. Better performance in video sequences was reported. Robust AAMs [80][54] extend the base formulations [99][38] to deal with occlusion (pixel outliers) by minimizing robust error functions (e.g. Hubber, Tukey bisquare or Cauchy functions) instead of the L2 norm, using Iteratively Reweighted Least Squares (IRLS). A robust norm AAM evaluation was presented in [101]. Fourier Active Appearance Models (FAAM) [82] presented an efficient joint alignment formulation, that fits an AAM across multiple filter responses (Gabor filters). Their work has shown substantial improvement in performance under illumination variation.

Extensions to 3D have also been proposed [31][8][7][6], with the 3D Morphable Model (3DMM) [105] one of the most popular. The 3DMMs shape model is built from 3D dense range scans, usually with several thousands of vertices and the appearance model consists of 3D cylindrical folded textures. Due to the large amount of data, the algorithm is quite slow taking minutes to fit the model to an image. Efficient 3DMMs, based on the IC algorithm, have also been proposed [90]. Still, its Jacobian and Hessian are only locally valid and take an average of 30s per frame, making it impracticable for real-time applications. Hybrid solutions, have also been proposed, such as the combined 2D+3D AAM [43][37] that uses 2D and 3D concepts working under a scaled orthographic projection model.

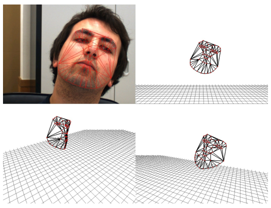
The AAMs have also been extended to 3D volumetric data (e.g. Magnetic Resonance Imaging - MRI or ultrasound imaging). Several methods have been proposed, mainly standard regression extensions [87][92][52] or IC based techniques [2][3].

Discriminative Methods: In this paradigm, both shape and appearance are combined by constraining an ensemble of local feature detectors to lie within the subspace spanned by a PDM. As discussed in section 1.2, fitting such a model requires a local search, using a feature detector, and a global optimization step that estimates the PDM parameters. The original Active Shape Models (ASM) [98], also known as Smart Snakes, uses as local detectors Gaussian models (mean and covariance) of grey level profile gradients (sampled along normal scanlines). The ASM global optimization consists of weighted peak responses taken from all landmarks (the matching was based in minimizing Mahalanobis distances between the sampled and the model profiles). The Pictorial Structures (PS) [63] introduced an efficient method of matching part-based models to images. The PS does not use an explicit shape model (like the PDM). The shape is encoded in a tree structure of geometric relationships between pairs of parts and the global set of final feature locations is efficiently found by dynamic programming. The PS is mainly a global search method and it has been shown less accurate when compared to specialized methods that use a full shape model [23]. The Boosted

Appearance Model (BAM) [108][109] uses a set of weak classifiers (Haar-like rectangular features) as an appearance model. These set of discriminative features are designed to distinguish between correct and incorrect image alignment (piecewise-affine warps, as the generative formulations). The classification boundary is learnt offline using a boosting framework (GentleBoost). The BAM model fitting consists of iteratively update the PDM parameters via gradient ascent such that the warped image achieves the maximal score from the trained classifier. The Constrained Local Model (CLM) [23][25], uses a joint model of shape and appearance, similar to the Active Appearance Model (AAM) [99]. However, the appearance of a CLM takes the form of rectangular regions surrounding individual features instead of triangulated patches covering the full face. The feature templates are then individually matched into the image using normalized correlation. The global step combines all the responses using the Nelder-Mead simplex algorithm. Boosted Regression Active Shape Models (BR-ASM) [24] describe the local landmark regions by a set of discriminative haar wavelets, learnt by means of GentleBoost. Both boosted classification and boosted regression strategies are evaluated (i.e. scoring the local alignment vs predicting the amount of local correction). Boosted regression as shown to have a wider range of convergence. In Convex Quadratic Fitting (CQF) [115] the local detectors are based on a linear Support Vector Machines (SVM) [106] built from aligned (positive) and misaligned (negative) patch examples. The response maps are found by performing an exhaustive local search around each current landmark estimate. The resulting responses are approximated by convex surfaces (full Gaussian functions) and the global optimization step estimates the PDM parameters that jointly combine all the Gaussians. Bayesian Constrained Local Models (BCLM) [104] generalizes the CQF (maximum likelihood) into a Bayesian formulation (MAP). In [46] a CQF similar approach is used, but the response maps are approximated by a Mixture of Gaussians. The work in [47] includes additional facial components detection to further constrain the PDM optimization. The component detection system enforces search directions that are determined by 'direction' classifiers (Adaboost using Gabor features with eight orientations and five scales). Recently, the Subspace Constrained Mean-Shift (SCMS) [41] use a nonparametric approximation of the response maps (using the mean-shift algorithm). However, in the global optimization the PDM parameters update is essentially a regularized projection of the mean-shift vector for each landmark, meaning that the optimization is very sensitive to outliers (when the mean-shift output is very far away from the correct landmark location).

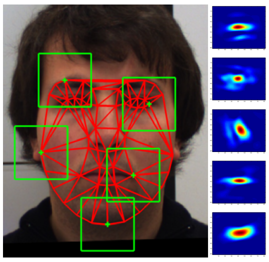
1.4 Thesis Overview

This thesis is structured in three main parts:



The first main part, presented in chapter 2, describes the **Generative 2.5D Active Appearance Models** (AAM) [70] methodology, an extension of the original 2D AAM [99][38] algorithm, extended to deal with a full perspective camera projection. The building process of both the 3D shape and the 2D appearance models are reported in detail, namely the linear 3D pose representation, the camera model, piecewise affine warping procedure, the Jacobian of The Warp and the initial estimate problem.

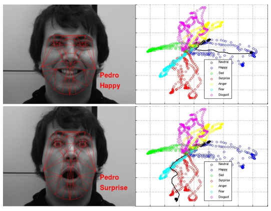
Several 2.5D AAM fitting strategies are proposed: the Simultaneous Forwards Additive (SFA), the Normalization Forwards Additive (NFA), their efficient approximations (ESFA and ENFA) and the robust to occlusions versions (RSFA, RNFA, ERSFA and ERSFA). Extensive results, involving the fitting convergence, fitting performance in unseen data, robustness to occlusion, tracking performance and pose estimation are shown, comparing the 2.5D AAM model with the standard 2D AAM [38] and the combined 2D+3D AAM [37] models.



The second main part, in chapter 3, entitled **Discriminative Bayesian Active Shape Models** (DBASM), describes the two discriminative algorithms developed. These approaches are closely related to Constrained Local Models (CLM) [23][25] and/or Active Shape Models (ASM) [93][98], where an ensemble of local feature detectors are constrained to lie within the subspace spanned by the shape model.

This chapter presents a new Bayesian global optimization strategy (DBASM [71]), where the posterior distribution of the shape parameters are inferred in a MAP sense by means of a Linear Dynamical System (LDS). The likelihood term is extracted from each one of the local response maps. Several local strategies used to represent these responses, either parametric and nonparametric, are described in detail. After, the second order global alignment is explained. It consists of an inference step that finds posterior distribution parameters, using a LDS.

Later, in section 3.5, the Bayesian Active Shape Models (BASM) [72] strategy is presented. BASM is an extension of the previous DBASM formulation where the prior distribution is explicitly modeled by means of recursive Bayesian estimation. Extensive results are presented, evaluating DBASM and BASM global optimization strategies, local face parts detectors and tracking performance in several standard datasets. Qualitative results taken from the challenging Labeled Faces in the Wild (LFW) dataset are also shown.



The third main part, in chapter 4, addresses the **Identity and Facial Expression Recognition** [66]. Face geometry is extracted from input images using the Active Appearance Models (AAM) and low dimensional manifolds were then derived using Laplacian EigenMaps (LE) resulting in two types of manifolds, one for model identity and the other for person-specific facial expression. The identity and facial expression recognition system, uses a two stage approach: First, a Support Vector Machines (SVM) [106] is used to establish identity across expression changes, while the second stage deals with person-specific expression recognition with a network of Hidden Markov Models (HMMs) [48]. Results taken from people exhibiting the six basic expressions (happiness, sadness, anger, fear, surprise, disgust) plus the neutral emotion are shown.

Finally, the chapter 5, contains a general **Conclusion and Discussion** of the overall work. As final note, some of the thesis related videos can be seen at the webpage: `http://www.isr.uc.pt/~pedromartins/Videos/PhD`.

1.5 Contributions

Briefly, the main contributions of this thesis are as follows:

1. Extension of the standard 2D Active Appearance Models (AAM) to deal with a full perspective projection model. The 2.5D AAM combines a 3D Point Distribution Model (PDM) and a 2D appearance model whose control points are defined by perspective projections of the PDM. The full six Degrees of Freedom (6 DOF) of the face are modeled by continuously integrate small pose changes at each frame since the beginning of tracking.

2. Fitting an AAM into a new image consists of optimizing the shape and appearance parameters that best describe the target face. Typically, a nonlinear optimization is involved, minimizing the difference between the synthetic appearance template and the and real appearance sampled from the image. In this thesis, we revisit the AAM model fitting strategies by carefully derive each new component for the 2.5D AAM case. Two main fitting algorithms, the Simultaneous Forwards Additive (SFA), the Normalization Forwards Additive (NFA) and their computationally efficient approximations are proposed. Robust SFA and NFA solutions, taking into account head partial and self occlusions are also proposed.

3. A novel discriminative face alignment technique is presented. The Discriminative Bayesian Active Shape Model (DBASM) is a new global optimization strategy that efficiently solves the global alignment. DBASM infers both the shape and the pose parameters, in a maximum a posteriori (MAP) sense, by means of a Linear Dynamical System (LDS). This approach models the covariance of the latent variables, i.e. it maintains 2^{nd} order statistics of the shape and pose parameters, which represents the confidence in the current parameters estimate.

4. A second Bayesian global optimization strategy (Bayesian Active Shape Models - BASM), an extension of DBASM, is also presented. BASM was designed to infer both the PDM and the pose parameters, in a MAP sense, by explicitly modelling the prior distribution (encoding the dynamic transitions of the PDM parameters).

Using recursive Bayesian estimation we model the prior distribution of the data as being Gaussian. The mean and covariance were assumed to be unknown and are treated as random variables.

5. Finally, the last part of the thesis, proposes a two step identity and facial expression recognition approach that relies a low dimensional representation of the geometry of the face. Face geometry is extracted from input images using the Active Appearance Models (AAM) and low dimensional manifolds were then derived using Laplacian EigenMaps (LE). The first stage uses a Support Vector Machines (SVM) to establish identity across expression changes. The second stage deals with person-specific facial expression recognition and is composed by a network of several Hidden Markov Models (HMM), each one specialized in a given facial emotion. The decision was made by the sequence that yielded the highest probability.

1.6 Publications

The main material in this thesis has been published in the following conference proceedings (listing by chapter):

Chapter 2 - Generative 2.5D Active Appearance Models:

- Face Alignment Through 2.5D Active Appearance Models [70]
  Pedro Martins, Rui Caseiro, Jorge Batista
  BMVC 2010 - British Machine Vision Conference

- Generative Face Alignment Through 2.5D Active Appearance Models
  Pedro Martins, Rui Caseiro, Jorge Batista
  CVIU - Computer Vision and Image Understanding [**Under Review - Minor Rev.**]

Chapter 3 - Discriminative Bayesian Active Shape Models:

- Let the Shape Speak - Discriminative Face Alignment using Conjugate Priors [72]
  Pedro Martins, Rui Caseiro, João F. Henriques, Jorge Batista
  BMVC 2012 - British Machine Vision Conference [**Oral Presentation**]

- Discriminative Bayesian Active Shape Models [71]

  Pedro Martins, Rui Caseiro, João F. Henriques, Jorge Batista

  ECCV 2012 - European Conference on Computer Vision

- Towards Generic Fitting Using Multiple Features Discriminative Active Appearance Models [69]

  Pedro Martins, Jorge Batista

  ICIP 2010 - IEEE International Conference on Image Processing

- Towards Generic Fitting Using Discriminative Active Appearance Models Embedded on a Riemannian Manifold [68]

  Pedro Martins, Jorge Batista

  VISAPP 2010 - International Conference on Computer Vision Theory and Applications

Chapter 4 - Identity and Facial Expression Recognition:

- Identity and Expression Recognition on Low Dimensional Manifolds [66]

  Pedro Martins, Jorge Batista

  ICIP 2009 - IEEE International Conference on Image Processing

- Simultaneous Identity and Expression Recognition Using Face Geometry on Low Dimensional Manifolds [67]

  Pedro Martins, Jorge Batista

  IbPria 2009 - Iberian Conference on Pattern Recognition and Image Analysis

Chapter 2

Generative 2.5D Active Appearance Models

This chapter addresses the matching of a 3D deformable face model to 2D images through a 2.5D Active Appearance Models (AAM). A 2.5D AAM that combines a 3D *metric* Point Distribution Model (PDM) and a 2D appearance model, whose control points are defined by a *full perspective* projection of the PDM, is presented. The advantage is that, assuming a calibrated camera, 3D metric shapes can be retrieved from single view images. Two model fitting algorithms and their computational efficient approximations are presented: the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA), both based on the Lucas-Kanade framework. The SFA algorithm searches for shape and appearance parameters simultaneously whereas the NFA projects out the appearance from the error image and searches only for the shape parameters (SFA is therefore more accurate). Robust solutions for the SFA and NFA are also described in order to take into account the self-occlusion or partial occlusion. Several performance evaluations for the SFA, NFA and theirs efficient approximations were performed. The experiments include evaluating the frequency of converge, the fitting performance in unseen data and the tracking performance in the FGNET Talking Face sequence. Results show that the 2.5D AAM can outperform both the 2D+3D combined models and the 2D standard methods. The robust extensions to occlusion were tested on synthetic sequences showing that the model can deal efficiently with large head rotation.

Publications

The contents of this chapter resulted in two main publications:

- Face Alignment Through 2.5D Active Appearance Models [70]
  Pedro Martins, Rui Caseiro, Jorge Batista
  BMVC 2010 - British Machine Vision Conference

- Generative Face Alignment Through 2.5D Active Appearance Models
  Pedro Martins, Rui Caseiro, Jorge Batista
  CVIU - Computer Vision and Image Understanding **[Under Review - Minor Rev.]**

2.1 Introduction

Facial image alignment is a key aspect in many computer vision applications, such as advanced human computer interaction, face recognition, head pose estimation, facial expression analysis, surveillance or realistic graphical animation. Detecting and tracking faces in video is a challenging task due to the non-rigidity structure of faces and also due to the large variability in shape, texture, pose and lighting conditions of their images.

The Active Appearance Model (AAM), introduced by [99], is one of the most effective face alignment technique with respect to fitting accuracy and efficiency. The standard AAMs are intrinsically 2D models, combining a 2D Point Distribution Model (PDM) [98][95] and a 2D appearance model into a single formulation using a fitting process that rely on a precomputed regression matrix. The AAM has been reformulated with true analytical derived gradients by Matthews *et al.*[38], achieving a better fitting accuracy and real-time performances using the Inverse Compositional (IC) [83] approach. Their solution is probably the fastest introduced so far, where its key to efficiency is that both the Jacobian and the Hessian matrices are constant and can be precomputed. A dual inverse compositional algorithm was also proposed in [4], dealing with both the geometric and photometric transformations in image registration under varying lighting conditions.

Although the excellent performance of the 2D AAM, its convergence ability is severely affected under large 3D head pose variations. To deal with this issue, several solutions have been proposed [100][73][30][13]. View-Based AAM [100] uses multiple 2D AAMs taken from each view, while issues related to self-occlusion are solved by using multiple view-specific templates. Similarly, the solution proposed by [73] uses multiple view appearance models although combined with a sparse 3D PDM. In [16] a IC algorithm for simultaneously fitting a 2D and a 3D PDM to multiple images is proposed. Their fitting methodology, instead of relying on multiple independent optimizations, is formulated in a single-objective optimization by enforcing the same 3D model across all the views. In [30][31], a 3D PDM derived from the Candide model [40] is used, being combined with a weak perspective model. In that work, head occlusions are handled by exploiting facial texture symmetry and the model fitting is based on a numerically estimated gradient.

Natural extensions to 3D have also been proposed [31][88][8][7][6], with the 3D Morphable Model (3DMM) [105] one of the most popular. There are several differences between AAMs and 3DMMs. The 3DMMs are built from 3D range scans, therefore are usually constructed to be denser, including several thousands of vertices whereas the AAMs use only a few tens. The appearance model consists of 3D cylindrical folded textures that are densely aligned between all samples in the training set. This huge alignment step involves a modified optical flow, designed to operate on cylindrical coordinates, and smooth interpolation methods to fill in the registration holes. A reflectance model (the Phong model) is also used, i.e. the appearance model also uses surface normals. The large amount of data, due to the density of the 3DMMs, makes the algorithm quite slow, requiring several minutes to fit per frame (50 minutes using a SGI R10000 processor). Efficient 3DMMs, working under a scaled orthographic projection model and based on the IC algorithm, have also been proposed [90]. Still, its Jacobian and Hessian are only locally valid and take an average of 30s per frame to fit, making it impracticable for real-time applications.

This chapter addresses the fitting of a 3D shape deformable face model from a single view through 2.5D AAM. The 2.5D model can be viewed as a 3D sparse PDM whose projections define 2D control points for the 2D appearance. This means that 2.5D data has components of both 2D image data and 3D volumetric shape data. Consequently, the 2.5D model combines the advantages of both 3DMMs and 2D AAMs, in particular the robustness to pose changes and the fitting speed. Face alignment on this 2.5D dimensional space will carry an extra level of complexity since the IC approach is invalid in this case [86]. To deal with this problem, Matthews *et al.*[37] proposed a 2D+3D AAM work around by exploiting the 2D and 3D shape models simultaneously. The shape instance generated by the 2D AAM is constrained to be consistent with the projection of a 3D *affine* shape (a 3D PDM is used, build from non rigid structure from motion [42]). This constraint is formulated as part of the cost function, where a balancing weight is added and the value of this weighting constant is determined manually. In [42] is also showed that any 2D projection of a 3D shape model can be represented by a 2D shape model but at the expense of using up to 6 times more parameters than using a 3D model. However, a weak perspective projection model was used in this demonstration and this property does not hold for the perspective projection model. The solution described in this chapter explores the advantages of using a single 3D model to constrain the possible 2D shape projection under the assumption of a full perspective model.

2.1.1 Contributions

The proposed solution extend the Active Appearance Model approach to deal with matching a 3D face shape model to a single 2D image using a perspective projection model, whereas previous approaches have generally only dealt with scaled orthographic projections. This approach uses a single 3D metric PDM combined with a full perspective model. The use of a full perspective model carries an important advantage over the state of the art solutions. Assuming a calibrated camera, an estimation of the 3D Euclidean shapes can be obtained from a single image and face tracking can be performed by using cameras with short focal length and strong radial distortion (e.g. a low cost webcam). Compared to [37], no balancing weight is required since the approach is based on a single, low dimensional, 3D PDM.

Two algorithms to fit a 3D deformable shape model to a 2D image are proposed. Both algorithms seek to minimize the difference between the projected model and the target image using slightly different strategies: The Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA), both based on the Lucas-Kanade forwards additive [85] update step. The SFA algorithm is computationally expensive but more accurate. It searches for shape and appearance parameters simultaneously whereas the NFA projects out the appearance from the error image and searches only for the shape parameters. Although both solutions require evaluating several components per iteration, efficient approximations are proposed leading to an efficient update step. By comparison, our fitting solution is based on analytically derived gradients (”true gradients”) rather than gradients approximated by numerical differences as in [30], genetic algorithms in [7] or generic optimization methods like the simplex in [8]. Finally, real-time performance can be achieving when using the efficient approximations, unlike the 3DMMs [105][90]. Moreover the methods used to acquired 3D dense shapes and textures normally demand very time consuming 3D reconstruction approaches or the use of expensive and cumbersome laser scan hardware.

Expanded solutions for the SFA and NFA are also proposed to handle self and partial occlusion, namely the Robust Simultaneous Forwards Additive (RSFA) and the Robust Normalization Forwards Additive (RNFA). These fitting methods use robust weighting functions that combine outlier estimation with pixel visibility extracted from the 3D pose.

In short, the main contributions in this chapter are as follows:

1. The use of a 2.5D AAM that combines a 3D metric Point Distribution Model (PDM) and a 2D appearance model whose control points are defined by full perspective projection of the PDM.

2. A unique shape model is used where all the six degrees of freedom (6 DOF) are modeled using a simple linear parametric model.

3. Two model fitting algorithms and their computationally efficient approximations are proposed: the Simultaneous Forwards Additive (SFA) and the Normalization

Forwards Additive (NFA).

4. Robust solutions for the SFA and NFA are also proposed in order to take into account head partial and self occlusions.

Other 2D AAM related extensions such as using Light-Invariant theory to deal with external shading [27], multi-band appearance models [53][60][39][81] or modifying the cost function in order to include the previously aligned frame as an additional constraint (SICOV) [111] can be easily incorporated into the proposed algorithms with expected improvements on the overall performance.

2.1.2 Outline

This chapter is organized as follows: **Section** 2.2 explains the 2.5D parametric model building process. The 3D PDM and 2D appearance models are both described in detail, as well as the full perspective camera model involved. **Section** 2.3 presents two model fitting algorithms, their respective efficient approximations and also the robust approaches to self and partial occlusion. In **Section** 2.4 is described how to efficiently evaluate the Jacobian of the warp for both shape and pose parameters, and the 2.5D AAM initial estimate problem is discussed in **Section** 2.5. Experimental results comparing both robust and non-robust fitting performances are presented in **Section** 2.6 and the results are discussed. Finally, **Section** 2.7 summarizes the chapter.

2.2 2.5D Parametric Models

The aim is to build a 2.5D AAM by combining a 3D metric Point Distribution Model (PDM) with a 2D appearance model whose control points are defined by full perspective projection of the PDM, as shown in figure 2.1. The 3D PDM is modeled by the shape and pose parameters, $\mathbf{p}$ and $\mathbf{q}$ respectively, that uniquely define a shape s in the 3D space whose projection into the image space sets 2D control points where the generated texture ($\boldsymbol{\lambda}$) is held.

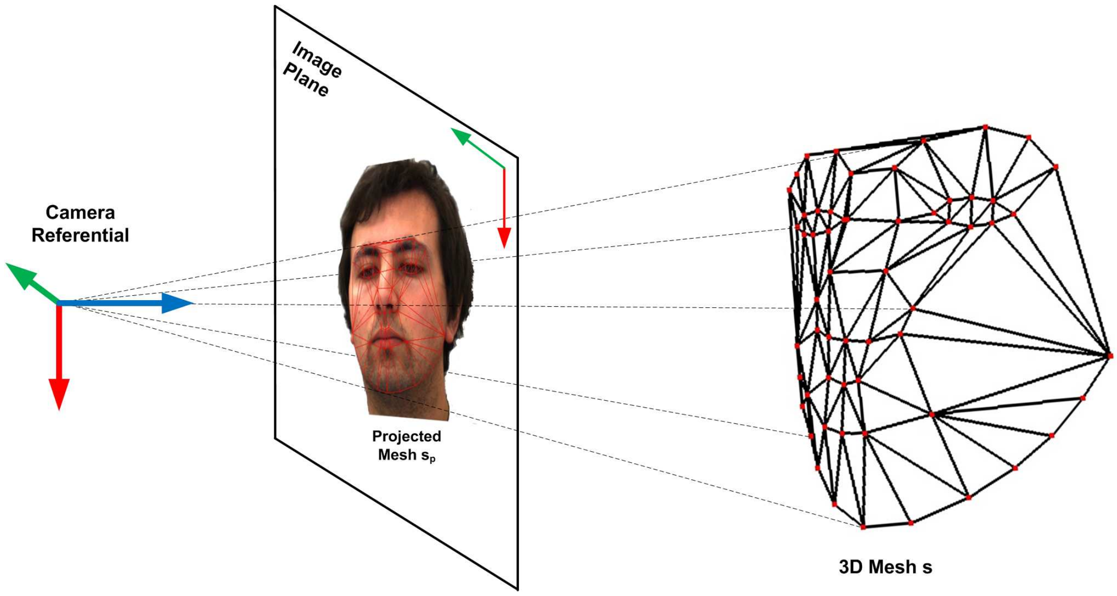
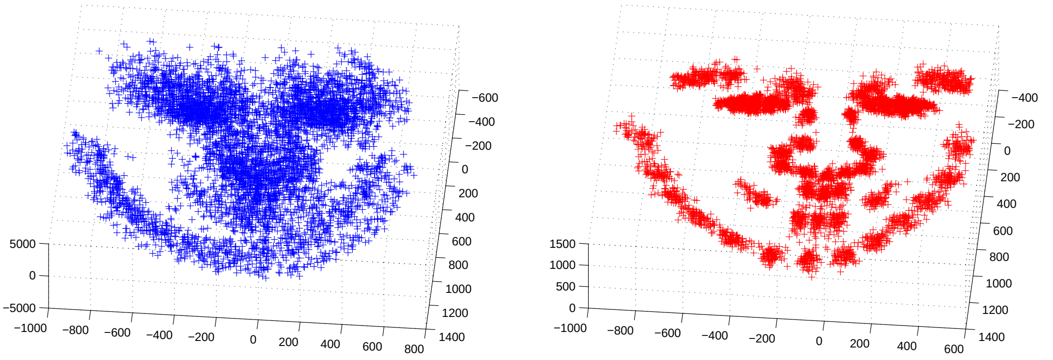


Figure 2.1: The 2.5D parametric model. The 3D shape model uniquely defines a shape in the 3D space whose projection into the image space sets the 2D control points to the generated texture by the appearance model.

2.2.1 The Shape Model

The shape of a non-rigid object can be expressed as a linear combination of a set of n basis shapes plus a rigid mean shape vector. This representation is also known as a Point Distribution Model (PDM) [95]. In PDM notation, each 3D v-point shape is defined by the vertex locations of a mesh $s = (X_1, \ldots, X_v, Y_1, \ldots, Y_v, Z_1, \ldots, Z_v)^T$ and the training data consists of a set of annotated images of those shapes (usually by hand). The shapes are then aligned into a common mean shape using a Generalized Procrustes Analysis (GPA) [15] that removes location, scale and rotation effects (fig. 2.2).

Applying a Principal Components Analysis (PCA) [56] to the aligned shapes, results the linear parametric model $s = s_0 + \Phi\mathbf{p}$, where $\mathbf{p}$ is a vector of shape configuration weights, s_0 is the mean shape (also refereed as the *base mesh*) and the basis $\Phi = [\phi_1 \cdots \phi_n]$ represent the allowed models of deformation. Figure 2.3 shows the visual representation of the first three modes of variation.



(a) Raw Data

(b) Aligned LandMarks

Figure 2.2: The shape alignment process using a Generalised Procrustes Analysis. a) Shows the raw shape data and b) the aligned data. All shapes are aligned into a common reference. The individual translation, scale and rotation effects are filtered.

In this work, the 3D PDM, including the full pose variation, is defined by

$$s = s_0 + \sum_{i=1}^{n} p_i\phi_i + \sum_{j=1}^{6} q_j\psi_j^{(t)} + \underbrace{\int_0^{t-1}\sum_{j=1}^{6} q_j\psi_j^{(t)}\partial t}_{s_\psi}\,. \tag{2.1}$$

where $\mathbf{p} = (p_1, \ldots, p_n)^T$ are the previous shape parameters, $\mathbf{q} = (q_1, \ldots, q_6)^T$ are the pose parameters and s_ψ is the contribution of pose increments over time t. The first two terms represent the PDM modes of deformation, the third term is the current estimated pose, and the last term (s_ψ) acts as an offset that accumulates pose increments from previous time frames. Note that $\psi_1, \ldots, \psi_6$ are a special set of eigenvectors that are only valid for small changes in pose. With this formulation, the shape model (eq.2.1) holds the full 6 DOF between the camera referential and the target face by means of incremental pose updates on the current mesh s.

Expressing a rotation of θ radians around an arbitrary axis $\mathbf{w} = (w_x, w_y, w_z)^T$ by the Rodrigues formula

$$\mathbf{R}(\mathbf{w}, \theta) = \mathbf{I}_3 + \hat{\mathbf{w}}\sin(\theta) + \hat{\mathbf{w}}^2(1 - \cos(\theta)), \quad \hat{\mathbf{w}} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}, \tag{2.2}$$

the incremental rotation update, based on the linearization of eq.2.2 and holding the first order terms, is given by

$$\mathbf{R}(\mathbf{w}, \theta) \approx \mathbf{I}_3 + \hat{\mathbf{w}}\theta. \tag{2.3}$$

By relaxing the constraint that $\mathbf{w}$ is of unit length, the θ coefficient can be dropped from eq.2.3. According, the pose update that transforms each 3D point $\mathbf{P}_i = (X_i, Y_i, Z_i)$ of the mesh s into $\mathbf{P}'_i$, is given by

$$\mathbf{P}'_i = \mathbf{R}(\mathbf{w})\mathbf{P}_i + \mathbf{T}_i \tag{2.4}$$

where $\mathbf{T}_i = (t_x, t_y, t_z)^T$ represents the 3D translation components. Defining the pose parameters vector as $\mathbf{q} = [w_x, w_y, w_z, t_x, t_y, t_z]^T$, eq.2.4 can be written as

$$\mathbf{P}'_i = \begin{bmatrix} 0 & Z_i & -Y_i & 1 & 0 & 0 \\ -Z_i & 0 & X_i & 0 & 1 & 0 \\ Y_i & -X_i & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{q}, \tag{2.5}$$

that describes how a single mesh point location is updated from $\mathbf{P}_i$ to $\mathbf{P}'_i$ through the pose vector $\mathbf{q}$.

Extending eq.2.5 to all the 3D mesh points of shape s, the small updates of the pose contribute to the current mesh through an amount of $\sum_{j=1}^{6} \psi_j q_j$. $\Psi = [\ \psi_1 \ \ldots \ \psi_6]$ is the extended version of eq.2.5, incorporating all the v points of the mesh s, and being expressed w.r.t. the *updated* base mesh (which is given by $s_0 + s_\psi$). It can be seen as a special set of *pose eigenvectors* and it is written as

$$\Psi(s_\psi) = \underbrace{\left[\begin{array}{cccccc} 0 & s_0^{z_1} + s_\psi^{z_1} & -s_0^{y_1} - s_\psi^{y_1} & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & s_0^{z_v} + s_\psi^{z_v} & -s_0^{y_v} - s_\psi^{y_v} & 1 & 0 & 0 \\ \hline -s_0^{z_1} - s_\psi^{z_1} & 0 & s_0^{x_1} + s_\psi^{x_1} & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -s_0^{z_v} - s_\psi^{z_v} & 0 & s_0^{x_v} + s_\psi^{x_v} & 0 & 1 & 0 \\ \hline s_0^{y_1} + s_\psi^{y_1} & -s_0^{x_1} - s_\psi^{x_1} & 0 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ s_0^{y_v} + s_\psi^{y_v} & -s_0^{x_v} - s_\psi^{x_v} & 0 & 0 & 0 & 1 \end{array}\right]}_{\psi_1,\ldots,\psi_6}. \tag{2.6}$$

Since Ψ is a function of s_ψ (since s_0 is constant), it requires being evaluated every time the mesh s is updated.

Finally, the last term of the PDM, s_ψ, consists in the integral form

$$s_\psi = \int_0^{t-1} \sum_{j=1}^{6} q_j \psi_j^{(t)} \partial t, \tag{2.7}$$

that collects small pose updates over time t. The s_ψ term plays a fundamental role. It overcomes the previous constraint on the incremental pose update so that the 6DOF can be successfully used and it allows updating the base mesh referential (as in eq.2.6) so that correct head rotations can be modeled.

2.2.2 The Camera Model

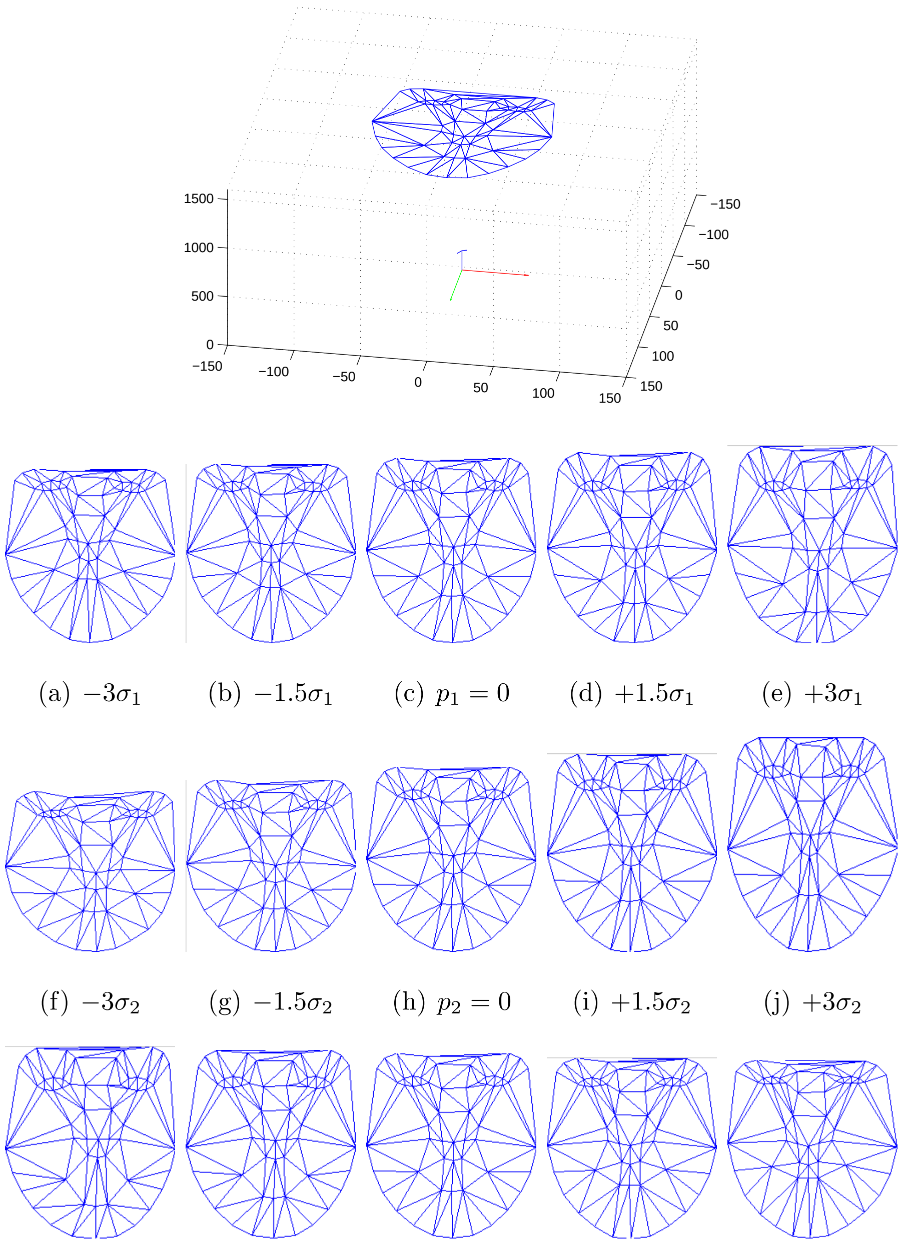
Using a *full perspective* camera, the 3D shape s generated by the PDM (eq.2.1), is projected into the image space as

$$\begin{bmatrix} w(x_1 \cdots x_v) \\ w(y_1 \cdots y_v) \\ w \cdots w \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & \alpha_s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\left[\begin{array}{c|c} \mathbf{R}_0 & \mathbf{t}_0 \end{array}\right]}_{\text{Base Pose}} \underbrace{\begin{bmatrix} s^{x_1} \cdots s^{x_v} \\ s^{y_1} \cdots s^{y_v} \\ s^{z_1} \cdots s^{z_v} \\ 1 \cdots 1 \end{bmatrix}}_{\text{PDM shape (eq.2.1)}} \tag{2.8}$$

where $\mathbf{K}$ is the camera matrix (with f_x, f_y the focal length, c_x, c_y the principal point and α_s the skew parameter) and it is assumed to be known. $\mathbf{R}_0$ and $\mathbf{t}_0$ are the rigid motion components between the camera frame and an extra referential where the PDM is defined. We define this rigid motion as the *base pose*. Both $\mathbf{R}_0$ and $\mathbf{t}_0$ are fixed and estimated during the PDM building process.

2.2.3 The Texture Model

The texture model is almost identical to the traditional 2D formulation [99], where each training image is texture-warped into a common frame using a warping function $\mathbf{W}$. This function $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ is a piecewise affine warp and is a function of the shape and pose parameters that define the 2D texture control points by means of the perspective



(a) $-3\sigma_1$ (b) $-1.5\sigma_1$ (c) $p_1 = 0$ (d) $+1.5\sigma_1$ (e) $+3\sigma_1$

(f) $-3\sigma_2$ (g) $-1.5\sigma_2$ (h) $p_2 = 0$ (i) $+1.5\sigma_2$ (j) $+3\sigma_2$

(k) $-3\sigma_3$ (l) $-1.5\sigma_3$ (m) $p_3 = 0$ (n) $+1.5\sigma_3$ (o) $+3\sigma_3$

Figure 2.3: The first three modes of variation of the 3D PDM. On top is shown the 3D base mesh $s_{0\mathbf{p}}$ and the camera frame. The PDM is composed by a mean shape plus a weighted eigenshape contribution. Each row of images shows the 2D image projection of how the shape deforms by spanning the weights $\mathbf{p}_i$ from $-3\sigma_i$ to $3\sigma_i$ $(i = 1, \ldots, n)$. The shape variances, σ_i^2 are captured when applying the PCA in the model building process. The middle column represents the mean shape projection $s_{0\mathbf{p}}$ when $\mathbf{p} = \mathbf{0}$.

projection of the mesh s (using eq.2.8). The warp is defined for all the projected pixels $\mathbf{x_p}$[1] contained within the *projected base mesh*, $s_{0\mathbf{p}}$, and is given by

$$\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}) = \mathbf{x}_{\mathbf{p}_i} + \alpha \left(\mathbf{x}_{\mathbf{p}_j} - \mathbf{x}_{\mathbf{p}_i} \right) + \beta \left(\mathbf{x}_{\mathbf{p}_k} - \mathbf{x}_{\mathbf{p}_i} \right), \forall \text{ triangles} \in s_{0\mathbf{p}} \tag{2.9}$$

where $\mathbf{x}_{\mathbf{p}_i}$, $\mathbf{x}_{\mathbf{p}_j}$, $\mathbf{x}_{\mathbf{p}_k}$ are triangle vertex's coordinates and α, β are the barycentric coordinates [12] for the pixel $\mathbf{x_p}$. The appearance model is obtained by applying a low memory PCA on all the warped training images and it is represented by a base appearance, $\mathbf{A}_0(\mathbf{x_p})$, plus a linear combination of m eigen images $\mathbf{A}_i(\mathbf{x_p})$, as

$$\mathbf{A}(\mathbf{x_p}) = \mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x_p}), \ \mathbf{x_p} \in s_{0\mathbf{p}} \tag{2.10}$$

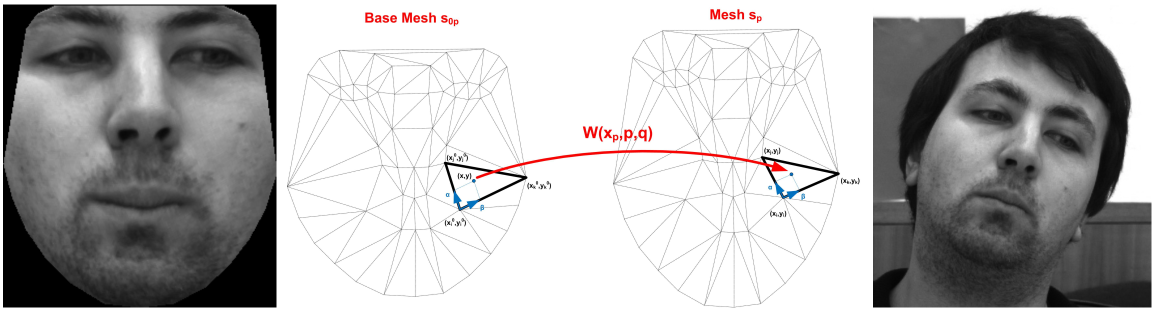
with λ_i being the appearance parameters. To model the gain and illumination offset effects, two extra appearance images are added $\mathbf{A}_{m+1}(\mathbf{x_p}) = \mathbf{A}_0(\mathbf{x_p})$ and $\mathbf{A}_{m+2}(\mathbf{x_p}) = \mathbf{1}$ which imposes the need for orthonormalization [85].

2.2.4 3D to 2D Piecewise Affine Warp

The warp function $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$, is a piecewise affine warp that is function of the shape and pose parameters. The warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ involves a 3D to 2D transformation, i.e. the 3D face mesh is generated from shape and pose parameters using eq.2.1 and then is projected into the image plane by a full perspective model using eq.2.8. As shown in figure 2.1, the converted 3D mesh points into 2D define the texture mapping control points. The piecewise affine warp is composed by sets of affine warps between corresponding triangles of the mesh. The base triangles are found by partitioning the convex hull of the projected mean shape, $s_{0\mathbf{p}}$, using the Delaunay triangulation, and each pixel belonging to a given triangle is mapped to its correspondent triangle using barycentric coordinates (see supplementary material section for details).

Figure 2.4 shows an illustration of this warping procedure. The warped image $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ is computed by backwards warping the input image $\mathbf{I}(\mathbf{x_p})$, therefore preventing holes, using the current estimate of the warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$. The warp is done for all the pixels $\mathbf{x_p}$ that lie within the projected base mesh $s_{0\mathbf{p}}$.

[1] During the remaining of the chapter, $\mathbf{x_p} = [x, y]^T$ defines a projected 3D point into the 2D image space, by eq.2.8.



(a) $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ (b) Piecewise affine warp illustration (c) Input image $\mathbf{I}(\mathbf{x_p})$

Figure 2.4: Piecewise affine warping. The warped image $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$ is computed by backwards warping the input image $\mathbf{I}(\mathbf{x_p})$, using the current estimate of the warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$.

2.3 Model Fitting

Fitting the AAM consists in finding the set of parameters, $\mathbf{p}$, $\mathbf{q}$ and $\boldsymbol{\lambda}$ that best describe the face in the target image. Since the Inverse Compositional (IC) approach [83] was proved in [86] to be invalid for the 2.5D case, two algorithms are proposed and described: the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA), both following the additive formulation proposed by Lucas-Kanade [84][11][64][20].

Both formulations include the 6DOF embedded in the PDM and just like the solutions initially proposed in [84][85], the SFA searches for all the parameters simultaneously whereas the NFA projects out the appearance from the error image. In section 2.3.3 it is shown how to maintain the fitting efficiency by making a simple approximation, precomputing a couple of terms. The experimental evaluation, as will be shown in section 2.6, proves that the proposed solution substantially improves the fitting performance.

2.3.1 Simultaneous Forwards Additive (SFA)

The SFA goal is to minimize the squared difference between the current instance of the appearance and the target warped image. The optimization consists in solving

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})) \right]^2 \tag{2.11}$$

simultaneously for the shape, pose and appearance parameters, $\mathbf{p}$, $\mathbf{q}$ and $\boldsymbol{\lambda}$ respectively. $\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$ represents the input image $\mathbf{I}(\mathbf{x_p})$ warped by $\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})$ as defined in section 2.2.3. The nonlinear optimization in eq.2.11 can be solved by gradient descent using additive updates to the parameters as

$$\sum_{\mathbf{x}\in s_{0\mathbf{p}}} [\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} (\lambda_i + \Delta\lambda_i)\mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p}+\Delta\mathbf{p},\mathbf{q}+\Delta\mathbf{q}))]^2. \tag{2.12}$$

Expanding and holding the first order Taylor terms gives[2]

$$\sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \Bigg[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) + \sum_{i=1}^{m+2} \Delta\lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})) \cdots$$
$$\cdots - \nabla\mathbf{I}\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{p}}\Delta\mathbf{p} - \nabla\mathbf{I}\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{q}}\Delta\mathbf{q}\Bigg]^2 \tag{2.13}$$

where $\nabla\mathbf{I}$ $\left(\nabla\mathbf{I} \equiv \nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})) = (\frac{\partial\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}{\partial x}, \frac{\partial\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}{\partial y})\right)$ represents the gradients of the image $\mathbf{I}(\mathbf{x_p})$ evaluated at $\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})$, before the warp. $\nabla\mathbf{I}$ is computed in the coordinate frame of $\mathbf{I}(\mathbf{x_p})$ and then warped back using the current warp estimate $\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})$. The terms $\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{p}}$ and $\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{q}}$ are Jacobians of the warp w.r.t. the shape and pose parameters, respectively[3].

Defining the combined parameters vector as $\mathbf{r} = [\mathbf{p}^T\ \mathbf{q}^T\ \boldsymbol{\lambda}^T]^T$ and denoting the $(n+6+m+2)$ Steepest Descent images $\mathbf{SD}(\mathbf{x_p})_{\mathsf{sfa}}$ as

$$\mathbf{SD}(\mathbf{x_p})_{\mathsf{sfa}} = \left[\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}_1} \ \dots\ \nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}_n}\ \nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{q}_1} \ \dots \nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{q}_6}\ -\mathbf{A}_1(\mathbf{x_p})\ \dots\ -\mathbf{A}_{m+2}(\mathbf{x_p}) \right], \tag{2.14}$$

[2] The derivation of eq.2.13 can be found in supplementary material section.

[3] From now on, $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$ and $\frac{\partial\mathbf{W}}{\partial\mathbf{q}}$ will be used as condensed representation for these Jacobians. Section 2.4 is totally dedicated to evaluate these Jacobians of the warp.

eq.2.13 can be written as

$$\sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) - \mathbf{SD}(\mathbf{x_p})_{\text{sfa}} \Delta \mathbf{r} \right]^2 . \tag{2.15}$$

Taking the partial derivative and making-it equal to zero $\left(\frac{\partial (\text{eq.2.15})}{\partial \Delta \mathbf{r}} = 0 \right)$ comes the closed from solution for the combined parameters update as

$$\Delta \mathbf{r} = \mathbf{H}_{\text{sfa}}^{-1} \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \mathbf{SD}(\mathbf{x_p})_{\text{sfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{sfa}} \tag{2.16}$$

where

$$\mathbf{H}_{\text{sfa}} = \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \mathbf{SD}(\mathbf{x_p})_{\text{sfa}}^T \mathbf{SD}(\mathbf{x_p})_{\text{sfa}} \tag{2.17}$$

represents the Gauss-Newton approximation to the Hessian matrix and $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$ represents the error image defined as

$$\mathbf{E}(\mathbf{x_p})_{\text{sfa}} = \mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})). \tag{2.18}$$

This procedure is done iteratively and the parameters are additively updated by $\mathbf{r} \leftarrow \mathbf{r} + \Delta \mathbf{r}$ until $\Delta \mathbf{r} \leq \varepsilon$ or a maximum number of iterations is reached.

The SFA is a computationally expensive algorithm since the reevaluation of the image warp $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$, the gradients before the warp $\nabla \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$, the error image $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$, the Jacobians $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$, $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$, that depend on $\mathbf{p}$ and $\mathbf{q}$ respectively, the $\mathbf{SD}(\mathbf{x_p})_{\text{sfa}}$ images and the Hessian matrix $\mathbf{H}_{\text{sfa}}$ and its inverse, are required for each iteration. This makes SFA algorithm rather slow but very accurate since it searches for shape, pose and appearance parameters simultaneously. Nevertheless, some components of the Jacobians $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$, $\frac{\partial \mathbf{W}}{\partial \mathbf{q}}$ are constant and can be precomputed (see section 2.4).

The algorithm 1 summarizes the SFA fitting method. Only at startup, a rough 3D pose estimation is required (the initial $\mathbf{q}$ parameters), taken from a combination of face detector (AdaBoost method [75]) and a 6DOF pose parameters extraction. See section 2.5 for details. The model starts with the initial shape parameters $\mathbf{p} = \mathbf{0}$ (the mean shape), $\boldsymbol{\lambda} = \mathbf{0}$ (the mean appearance) and $s_\psi = \mathbf{0}$ (zero pose offset).

```
Precompute:
The 2.5D parametric models: $(s_0,\ \Phi,\ \Psi)$ and $(\mathbf{A}_0(\mathbf{x_p}), \mathbf{A}_i(\mathbf{x_p}))$
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ for $k = 1, \ldots, v$ (see figure 2.8)
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp image $\mathbf{I}(\mathbf{x_p})$ with $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$, computing $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate the gradients $\nabla\mathbf{I}(\mathbf{x_p})$ and warp to $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Compute the Error image $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$ using eq.2.18
    Evaluate the Jacobian of the warp w.r.t shape $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Evaluate the Jacobian of the warp w.r.t pose $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute Steepest Descent images $\mathbf{SD}(\mathbf{x_p})_{\text{sfa}}$ using eq.2.14
    Find the Hessian matrix $\mathbf{H}_{\text{sfa}}$ and its inverse with eq.2.17
    Compute the parameters updates $\Delta\mathbf{r}$ with eq.2.16
    Update parameters $\mathbf{r} \leftarrow \mathbf{r} + \Delta\mathbf{r}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $||\Delta r|| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 1: Simultaneous Forwards Additive (SFA).

2.3.2 Normalization Forwards Additive (NFA)

A slightly different algorithm that minimizes the expression in eq.2.11 is the NFA algorithm. An alternative way of dealing with the linear appearance variation is to project out the appearance images $\mathbf{A}_i(\mathbf{x_p})$ from the error image [85]. Denoting the appearance into a single image by

$$\mathbf{A}(\mathbf{x_p}, \boldsymbol{\lambda}) = \mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}), \tag{2.19}$$

eq.2.11 can be written as

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \left[\mathbf{A}(\mathbf{x_p}, \boldsymbol{\lambda}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))\right]^2 . \tag{2.20}$$

Supposing, by now, that there is no appearance variation, which means that $\mathbf{A}(\mathbf{x_p}, \boldsymbol{\lambda}) = \mathbf{A}_0(\mathbf{x_p})$, the $(n+6)$ modified $\mathbf{SD}_{\text{nfa}}(\mathbf{x_p})$ images are represented, as

$$\mathbf{SD}(\mathbf{x_p})_{\text{nfa}} = \left[\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}_1} \;\ldots\; \nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}_n} \;\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{q}_1} \ldots \nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{q}_6}\right]. \tag{2.21}$$

Applying a first order Taylor expansion to eq.2.20 results

$$\sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) - \mathbf{SD}_{\text{nfa}}(\mathbf{x_p}) \begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix}\right]^2 \tag{2.22}$$

and following the same strategy used for the SFA approach, the error image and the Hessian are, respectively, given by

$$\mathbf{E}(\mathbf{x_p})_{\text{lk}} = \mathbf{A}_0(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) \tag{2.23}$$

and

$$\mathbf{H}_{\text{nfa}} = \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \mathbf{SD}(\mathbf{x_p})^T_{\text{nfa}} \mathbf{SD}(\mathbf{x_p})_{\text{nfa}}. \tag{2.24}$$

Dealing with the full appearance variation $(\mathbf{A}(\mathbf{x_p}, \boldsymbol{\lambda}))$ requires a *normalization* procedure. It is accomplished in the following two steps:

(1) Project the error image, $\mathbf{E}(\mathbf{x})_{\text{lk}}$, into the appearance basis by estimating the $m+2$ appearance parameters using

$$\lambda_i = \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \mathbf{A}_i(\mathbf{x_p})\mathbf{E}(\mathbf{x_p})_{\text{lk}}, \quad i = 1, \ldots, m+2 \tag{2.25}$$

(2) Remove the component of the error image in the direction of $\mathbf{A}_i(\mathbf{x_p})$ finding the normalized error image

$$\mathbf{E}_{\text{nfa}}(\mathbf{x_p}) = \mathbf{E}(\mathbf{x_p})_{\text{lk}} - \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}). \tag{2.26}$$

The NFA method consists in normalizing the error image (that has appearance $\mathbf{A}(\mathbf{x_p}, \boldsymbol{\lambda})$) so that the component of the error image in the direction $\mathbf{A}_i(\mathbf{x_p})$ is zero. This step has the advantage of estimate the appearance parameters $\boldsymbol{\lambda}$. Finally, the parameters updates are given by

$$\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix} = \mathbf{H}_{\text{nfa}}^{-1} \sum_{\mathbf{x_p} \in s_{0\mathbf{p}}} \mathbf{SD}(\mathbf{x_p})_{\text{nfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{nfa}}. \tag{2.27}$$

The NFA algorithm is less computationally expensive than the SFA, since it projects out the appearance from the error image and searches only for the shape and pose parameters. As shown in algorithm 2, each iteration requires reevaluating the image warp $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$, the warped gradients $\nabla\mathbf{I}$, the error image, $\mathbf{E}(\mathbf{x_p})_{\text{lk}}$, the normalized error image $\mathbf{E}(\mathbf{x_p})_{\text{nfa}}$, the Jacobians $\frac{\partial\mathbf{W}}{\partial\mathbf{p}}$, $\frac{\partial\mathbf{W}}{\partial\mathbf{q}}$, $\mathbf{SD}(\mathbf{x_p})_{\text{nfa}}$ and the Hessian $\mathbf{H}_{\text{nfa}}^{-1}$. However, note that the $\mathbf{SD}(\mathbf{x_p})_{\text{nfa}}$ images are much smaller in number than the $\mathbf{SD}(\mathbf{x_p})_{\text{sfa}}$, i.e. $(n << m)$, with typical values of n about $10 - 20$ and m about $50 - 80$. The NFA algorithm performs much faster than the SFA.

2.3.3 Efficient Approximations to SFA and NFA

Some computational load can be reduced by eliminating the need to recompute the image gradients at each iteration. Following the idea proposed by Hager *et al.*[34], and assuming existence of good estimates for all the parameters $\mathbf{p}$, $\mathbf{q}$ and $\boldsymbol{\lambda}$ (in eq.2.11), the error image $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$ will be $\approx \mathbf{0}$ and we can say that:

$$\begin{aligned} \left(\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p})\right) &\approx \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) \\ &\Downarrow \\ \underbrace{\left(\nabla\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \nabla\mathbf{A}_i(\mathbf{x_p})\right)}_{\nabla\mathbf{A}_i(\mathbf{x_p}, \boldsymbol{\lambda})} &\approx \nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})). \end{aligned} \tag{2.28}$$

```
Precompute:
The 2.5D parametric models: $(s_0,\ \Phi,\ \Psi)$ and $(\mathbf{A}_0(\mathbf{x_p}), \mathbf{A}_i(\mathbf{x_p}))$
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ for $k = 1, \ldots, v$ (see figure 2.8)
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}$ with $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$, computing $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate the gradients $\nabla\mathbf{I}(\mathbf{x_p})$ and warp to $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Compute the Error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{lk}}$, eq.2.23
    Project-out the error image into $\mathbf{A}_i(\mathbf{x_p})$ basis and estimate the appearance
    parameters $\boldsymbol{\lambda}$ using eq.2.25
    Find the normalization error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{nfa}}$ with eq.2.26
    Evaluate the Jacobian of the warp w.r.t shape $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Evaluate the Jacobian of the warp w.r.t pose $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute Steepest Descent images $\mathbf{SD}(\mathbf{x_p})_{\mathrm{nfa}}$ using eq.2.21
    Find the Hessian matrix $\mathbf{H}_{\mathrm{nfa}}$ and its inverse
    Compute the parameters updates $\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix}$ with eq.2.27
    Update parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ and $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $\left\| \begin{matrix} \Delta p \\ \Delta q \end{matrix} \right\| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 2: Normalization Forwards Additive (NFA).

Under this approximation, the Efficient SFA/NFA Steepest Descent images from eq.2.14 and eq.2.21, respectively, can be rewritten as

$$\mathbf{SD}(\mathbf{x_p})_{\text{esfa}} = \Big[\nabla\mathbf{A}_i(\mathbf{x_p},\boldsymbol{\lambda})\frac{\partial\mathbf{W}}{\partial\mathbf{p}_1}\ \ldots\ \nabla\mathbf{A}_i(\mathbf{x_p},\boldsymbol{\lambda})\frac{\partial\mathbf{W}}{\partial\mathbf{p}_n}\ \nabla\mathbf{A}_i(\mathbf{x_p},\boldsymbol{\lambda})\frac{\partial\mathbf{W}}{\partial\mathbf{q}_1}\ldots$$
$$\ldots\nabla\mathbf{A}_i(\mathbf{x_p},\boldsymbol{\lambda})\frac{\partial\mathbf{W}}{\partial\mathbf{q}_6}\ \ -\mathbf{A}_1(\mathbf{x_p})\ \ldots\ -\mathbf{A}_{m+2}(\mathbf{x_p})\Big], \quad (2.29)$$

and

$$\mathbf{SD}(\mathbf{x_p})_{\text{enfa}} = \Big[\nabla\mathbf{A}_0(\mathbf{x_p})\frac{\partial\mathbf{W}}{\partial\mathbf{p}_1}\ \ldots\ \nabla\mathbf{A}_0(\mathbf{x_p})\frac{\partial\mathbf{W}}{\partial\mathbf{p}_n}\ \nabla\mathbf{A}_0(\mathbf{x_p})\frac{\partial\mathbf{W}}{\partial\mathbf{q}_1}\ldots\nabla\mathbf{A}_0(\mathbf{x_p})\frac{\partial\mathbf{W}}{\partial\mathbf{q}_6}\Big]. \quad (2.30)$$

The approximation in eq.2.28, besides providing extra computation efficiency (the gradients of the template $\nabla\mathbf{A}_0$ can be precomputed when using ENFA and also the gradients of all the eigen faces $\nabla\mathbf{A}_i$ when using ESFA), it has the great advantage of providing better stability to noise sensitivity since it avoids the reevaluation of the gradients in the input image $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$ and at both warps $\frac{\partial\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}{\partial x}$, $\frac{\partial\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}{\partial y}$ of each iteration.

Figure 2.5 shows an example of the ESFA fitting method applied in a video sequence. Each image shows three different views of the 3D mesh and input frame overlaid with its current projection.

The algorithms 10 and 11, shown in A.2, summarize the detailed steps of the Efficient versions of the Simultaneous and the Normalization Forwards Additive approaches.

2.3.4 Robust Fitting

Both SFA and NFA are data driven algorithms and the error image continuously drives the models in further updates. In the case of occlusion, the error image accounts for all the pixels equally (L2 norm) leading the model to diverge. To overcome this problem, occlusion can be modeled as outlier pixels in the appearance model and handled by robust fitting methods [80] [54], namely by Iteratively Reweighted Least Squares (IRLS) where outliers are not accounted for the parameters updates.

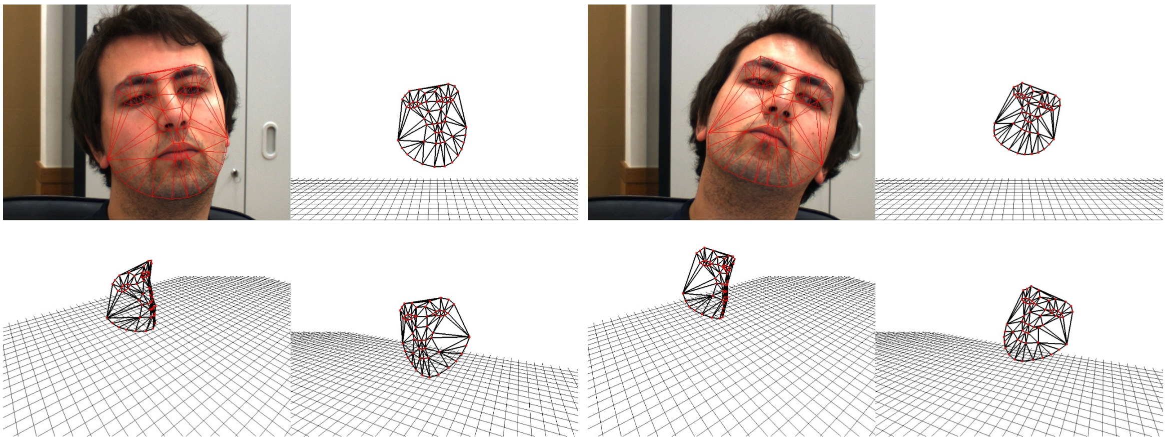


Figure 2.5: 2.5D AAM fitting using the Efficient Simultaneous Forwards Additive (ESFA) algorithm. Each image shows the input frame overlaid with the projected mesh and three different views of the current 3D mesh s. The full video sequence can be seen at http://www.isr.uc.pt/~pedromartins/Videos/PhD.

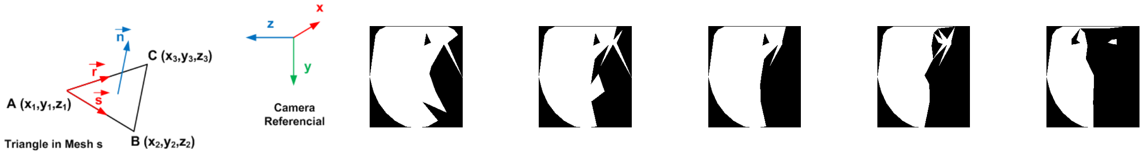
The robust fitting seeks to minimize

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \rho\left(\left[\underbrace{\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}_{\mathbf{E}(\mathbf{x_p})_{\text{sfa}}}\right]^2, \sigma_{\mathbf{x_p}}\right) \tag{2.31}$$

where $\rho(.)$ is a robust error function that has the purpose of weighting the large errors on $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$ so that they have less significance in updating the fitting parameters. The vector of scale parameters is defined as $\sigma_{\mathbf{x_p}}$ and can be estimated from the error image, $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$. The notation $\sigma_{\mathbf{x_p}}$ reflects that each pixel $\mathbf{x_p}$ is treated independently i.e. the decision if a pixel is occluded is not influenced by any other pixel.

Modified Robust Error Function

Several robust error functions can be used, such as the Hubber, the Tukey or the Cauchy function (see [101] for an AAM related comparison). In this work a slightly modified robust error function, based on the Talwar function, is used. The Talwar



(a) Back-face Culling (b) 40° (c) 50° (d) 60° (e) 75° (f) 90°

Figure 2.6: a) Back-face Culling illustration. $\mathbf{n}$ is the normal vector from a triangle in mesh s and $\mathbf{z}$ is the view vector from the camera reference. Images (b)(c)(d)(e) show the triangle visibility mask over the projected base mesh, $s_{0\mathbf{p}}$, for a head pan variation of 40°, 50°, 60°, 75° and 90° w.r.t the base pose using the Back-face Culling technique. Non-visible triangles (in black) are not used to update the parameters.

function assigns a weight of 1 to inliers and 0 to outliers, according to

$$\rho(\mathbf{E}(\mathbf{x_p}), \sigma_{\mathbf{x_p}}) = \begin{cases} 1, & |\mathbf{E}(\mathbf{x_p})| \leq \sigma_{\mathbf{x_p}} \\ 0, & |\mathbf{E}(\mathbf{x_p})| > \sigma_{\mathbf{x_p}}. \end{cases} \tag{2.32}$$

The scale parameter, $\sigma_{\mathbf{x_p}}$, can also be estimated from several ways. Since statistical distribution of the error image is unknown it can be assumed that the error image has a given percentage of outliers (e.g. 5% or 10%) and $\sigma_{\mathbf{x_p}}$ is set such that the largest user defined percentage of error pixels are rejected. Other solution, consists in estimate $\sigma_{\mathbf{x_p}}$ from the fitting error residuals using the Median of Absolute Deviations (MAD). The scale estimation can be moved into the AAM model building process by simply running, in an offline mode, a fitting algorithm for every (unoccluded) training image and then estimate the MAD fitting error. Figure 2.7 shows robust fitting results using the MAD as an estimate to the scale parameters $\sigma_{\mathbf{x_p}}$.

The 2.5D proposed model has the advantage of being able to estimate the visible areas (say mesh triangles) in the image projection model. The robust function modification consists in using information about the triangles visibility over the projected base mesh (by Back-face Culling) and select the invisible triangles by the camera to be dropped. These occluded triangles are established as outliers and are not taken into consideration in the fitting process. See figure 2.6.

Robust Fitting Algorithms (RSFA and RNFA)

The derivation of the Robust versions of SFA and NFA algorithms, RSFA and RNFA respectively, is similar to those of section 2.3, where the RSFA final parameters update is given by

$$\Delta\mathbf{r} = \mathbf{H}_{\text{rsfa}}^{-1} \sum_{\mathbf{x} \in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})_{\text{sfa}}^2)\mathbf{SD}(\mathbf{x_p})_{\text{sfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{sfa}} \tag{2.33}$$

being $\rho(\mathbf{E}(\mathbf{x_p})_{\text{sfa}}^2)$ a weight mask that measures the confidence of each pixel over the base mesh. The Hessian is defined as

$$\mathbf{H}_{\text{rsfa}} = \sum_{\mathbf{x} \in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})_{\text{sfa}}^2)\mathbf{SD}(\mathbf{x_p})_{\text{sfa}}^T \mathbf{SD}(\mathbf{x_p})_{\text{sfa}}. \tag{2.34}$$

Algorithm 3 describes in detail the steps required for the RSFA.

In the same way, the Robust version of NFA (RNFA) includes a weight mask in the Steepest Descent images, when evaluating the Hessian matrix,

$$\mathbf{H}_{\text{rnfa}} = \sum_{\mathbf{x} \in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})_{\text{rnfa}}^2)\mathbf{SD}(\mathbf{x_p})_{\text{nfa}}^T \mathbf{SD}(\mathbf{x_p})_{\text{nfa}} \tag{2.35}$$

and the parameters updates become

$$\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix} = \mathbf{H}_{\text{rnfa}}^{-1} \sum_{\mathbf{x} \in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})_{\text{rnfa}}^2)\mathbf{SD}(\mathbf{x_p})_{\text{nfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{rnfa}} \tag{2.36}$$

with the error image being

$$\mathbf{E}(\mathbf{x_p})_{\text{rnfa}} = \mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})). \tag{2.37}$$

Just like in the NFA algorithm, the RNFA requires an appearance normalization step for the error image. As referred in section 2.3.2, the goal of this normalization step is to make the component of the error image in the direction of $\mathbf{A}_i(\mathbf{x_p})$ to be zero. The NFA method deals with this by simply projecting the error image into the appearance basis ($\mathbf{A}_i(\mathbf{x_p})$). However the same approach can not be used in the robust version. With the use of a robust error function, $\rho(.)$, the appearance vectors are no longer orthonormal.

```
Precompute:
The 2.5D parametric models: $(s_0,\ \Phi,\ \Psi)$ and $(\mathbf{A}_0(\mathbf{x_p}), \mathbf{A}_i(\mathbf{x_p}))$
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ for $k = 1, \ldots, v$ (see figure 2.8)
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}$ with $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$, computing $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate the gradients $\nabla\mathbf{I}(\mathbf{x_p})$ and warp to $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate triangle visibility by Back Face Culling
    Compute the Error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{sfa}}$ using eq.2.18
    Estimate the weight mask $\rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{sfa}})$
    Evaluate the Jacobian of the warp w.r.t shape $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Evaluate the Jacobian of the warp w.r.t pose $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute Steepest Descent images $\mathbf{SD}(\mathbf{x_p})_{\mathrm{sfa}}$ using eq.2.14
    Find the Hessian matrix $\mathbf{H}_{\mathrm{rsfa}}$ and its inverse with eq.2.34
    Compute the parameters updates, $\Delta\mathbf{r}$, with eq.2.33
    Update parameters $\mathbf{r} \leftarrow \mathbf{r} + \Delta\mathbf{r}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $||\Delta \boldsymbol{r}|| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 3: Robust Simultaneous Forwards Additive (RSFA).

A slightly modified solution of the normalization step, initially proposed in [85], can be used. Starting from the error image $\mathbf{E}(\mathbf{x})_{\text{nfa}}$, the goal is to compute the appearance parameters update $\Delta\boldsymbol{\lambda}$ that minimize

$$\sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\text{rnfa}}) \left(\mathbf{E}_{\text{rnfa}}(\mathbf{x_p}) + \sum_{i=1}^{m+2} \Delta\lambda_i \mathbf{A}_i(\mathbf{x_p})\right)^2, \tag{2.38}$$

which has the least squares minimum given by

$$\Delta\boldsymbol{\lambda} = \mathbf{H}_{\text{A}}^{-1} \sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\text{rnfa}}) \mathbf{A}_i(\mathbf{x_p})^T \mathbf{E}(\mathbf{x_p})_{\text{rnfa}} \tag{2.39}$$

where

$$\mathbf{H}_{\text{A}} = \sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\text{rnfa}}) \sum_{i=1}^{m+2} \mathbf{A}_i(\mathbf{x_p})^T \mathbf{A}_i(\mathbf{x_p}) \tag{2.40}$$

is the appearance Hessian.

Algorithm 4 describe the RNFA algorithm steps, including the robust appearance normalization.

Efficient Robust Approximations (ERSFA and ERNFA)

The efficient approximations presented in section 2.3.3 are also valid for the robust fitting versions. The main changes w.r.t. the standard versions (RSFA and RNFA) are the use of efficient Steepest Descent images in eqs.2.29 and 2.30, respectively. See algorithms 12 and 13 in A.2 for details. Figure 2.7-top shows some occlusion robust examples using the ERNFA algorithm in a video sequence. Dealing with self-occlusion effects can be seen in figure 2.7-bottom where the ERSFA algorithm was used.

2.4 The Jacobian of The Warp

The Jacobians of the warp measure the rate of change of the destination in the warp $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ w.r.t. the parameters $\mathbf{p}$ and $\mathbf{q}$. Two Jacobians must be derived, $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$, w.r.t. shape and pose parameters, respectively.

```
Precompute:
The 2.5D parametric models: $(s_0, \Phi, \Psi)$ and $(\mathbf{A}_0(\mathbf{x_p}), \mathbf{A}_i(\mathbf{x_p}))$
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ for $k = 1, \ldots, v$ (see figure 2.8)
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}$ with $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$, computing $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate the gradients $\nabla\mathbf{I}(\mathbf{x_p})$ and warp to $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
    Evaluate triangle visibility by Back Face Culling
    Compute the Error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{rnfa}}$ using eq.2.37
    Estimate the weight mask $\rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{rnfa}})$
    Find the Hessian appearance $\mathbf{H}_{\mathrm{A}}$ with eq.2.40
    Compute the appearance parameters update $\Delta\boldsymbol{\lambda}$ with eq.2.39
    Update appearance parameters $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$
    Recompute $\mathbf{E}(\mathbf{x_p})_{\mathrm{rnfa}}$ using eq.2.37 (normalized error image)
    Evaluate the Jacobian of the warp w.r.t shape $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Evaluate the Jacobian of the warp w.r.t pose $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute Steepest Descent images $\mathbf{SD}(\mathbf{x_p})_{\mathrm{nfa}}$ using eq.2.21
    Find the Hessian matrix $\mathbf{H}_{\mathrm{rnfa}}$ and its inverse with eq.2.35
    Compute the parameters updates $\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix}$ with eq.2.36
    Update parameters $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ and $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $\left\| \begin{matrix} \Delta \boldsymbol{p} \\ \Delta \boldsymbol{q} \end{matrix} \right\| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 4: Robust Normalization Forwards Additive (RNFA).

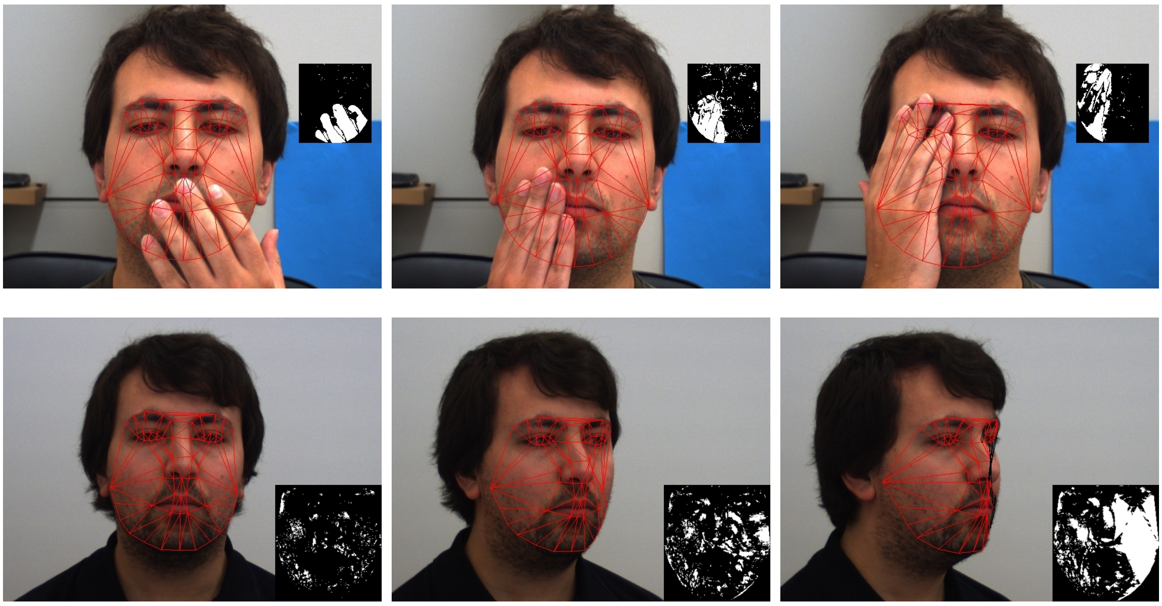


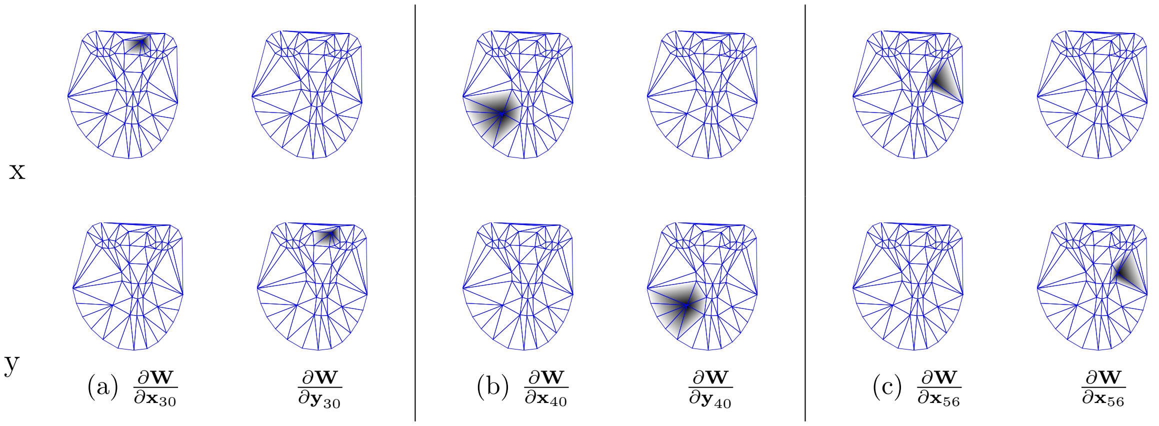
Figure 2.7: The top images show the robust 2.5D AAM fitting using the ERNFA algorithm. The weight mask $\rho(\mathbf{E}(\mathbf{x_p})^2_{\text{rnfa}})$ is shown on the right. The scale parameters $\sigma_{\mathbf{x_p}}$ were estimated assuming that there always exists 10% of outliers in the error image. On bottom images the ERSFA algorithm was used with $\sigma_{\mathbf{x_p}}$ estimated from the fitting error MAD. Both full video sequences can be seen at http://www.isr.uc.pt/~pedromartins/Videos/PhD.

2.4.1 Jacobian of The Warp for The Shape Parameters

The Jacobian of the warp for the shape parameters can be decomposed by the chain rule as

$$\frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}} = \sum_{k=1}^{v} \left[\frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{x}_k} \frac{\partial \mathbf{x}_k}{\partial \mathbf{p}} + \frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{y}_k} \frac{\partial \mathbf{y}_k}{\partial \mathbf{p}} \right]. \tag{2.41}$$

Taking eq.2.9, comes that $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{x}_k} = (1-\alpha-\beta, 0)$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{y}_k} = (0, 1-\alpha-\beta)$. These Jacobians are images w.r.t. a particular vertex and have the same size of the projected base mesh $s_{0\mathbf{p}}$. Figure 2.8 shows examples of these images for some landmarks (note the x and y components). The Jacobians are only non zero around the neighbor triangles of vertex k^{th}, taking the maximum value of 1 at the vertex location and decaying linearly with a rate of $(1-\alpha-\beta)$ to the other surrounding vertex's.



(a) $\frac{\partial \mathbf{W}}{\partial \mathbf{x}_{30}}$ $\frac{\partial \mathbf{W}}{\partial \mathbf{y}_{30}}$ (b) $\frac{\partial \mathbf{W}}{\partial \mathbf{x}_{40}}$ $\frac{\partial \mathbf{W}}{\partial \mathbf{y}_{40}}$ (c) $\frac{\partial \mathbf{W}}{\partial \mathbf{x}_{56}}$ $\frac{\partial \mathbf{W}}{\partial \mathbf{x}_{56}}$

Figure 2.8: (a) (b) (c) Shows $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{y}_k}$ for the landmarks 30, 40 and 56, respectively. Top and bottom rows represent $\mathbf{W_x}(\mathbf{x_p},\mathbf{p},\mathbf{q})$ and $\mathbf{W_y}(\mathbf{x_p},\mathbf{p},\mathbf{q})$ components. For clarity the shown images are black/white inverted. The location of the vertex has a maximum value and decays linearly to its neighbors. Note the highly sparse matrices shown.

The remaining terms $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}_i}$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}_i}$ are both scalars, found by combining eq.2.8 and eq.2.1, as

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \underbrace{\mathbf{K}\left[\begin{array}{c|c} \mathbf{R}_0 & \mathbf{t}_0 \end{array}\right]}_{\mathbf{M}_0} \begin{bmatrix} s_0^{x_k} + p_i\phi_i^{x_k} + \sum_{j\neq i}^{n} p_j\phi_j^{x_k} + \sum_{j=1}^{6} q_j\Psi_j^{x_k} + s_\psi^{x_k} \\ s_0^{y_k} + p_i\phi_i^{y_k} + \sum_{j\neq i}^{n} p_j\phi_j^{y_k} + \sum_{j=1}^{6} q_j\Psi_j^{y_k} + s_\psi^{y_k} \\ s_0^{z_k} + p_i\phi_i^{z_k} + \sum_{j\neq i}^{n} p_j\phi_j^{z_k} + \sum_{j=1}^{6} q_j\Psi_j^{z_k} + s_\psi^{z_k} \\ 1 \end{bmatrix}. \quad (2.42)$$

To compute $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}_i}$ we take the differential $\frac{\partial}{\partial \mathbf{p}_i}(\frac{w\mathbf{x}_k}{w})$ from eq.2.42, and do the same for $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}_i} = \frac{\partial}{\partial \mathbf{p}_i}(\frac{w\mathbf{y}_k}{w})$, resulting in

$$\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}_i} = \frac{\xi_1\Xi_3 - \Xi_1\xi_3}{(\Xi_3)^2} \quad \text{and} \quad \frac{\partial \mathbf{y}_k}{\partial \mathbf{p}_i} = \frac{\xi_2\Xi_3 - \Xi_2\xi_3}{(\Xi_3)^2} \quad (2.43)$$

with $i = 1, \ldots, n$ (shape parameters) and $k = 1, \ldots, v$ (landmarks). The $\xi_1, \xi_2, \xi_3, \Xi_1, \Xi_2$ and Ξ_3 are all scalars values defined in A.1. Note that the amount $\sum_{j\neq i}^{n} p_j\phi_j$ (non-rigid shape deformation excluding the i^{th} parameter) is constant when taking the i^{th} shape parameter differential.

As previously mentioned, $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}_i}$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}_i}$ are both scalars and depend on $\mathbf{p}$ and $\mathbf{q}$ by means of Ξ_1, Ξ_2 and Ξ_3. Reevaluating the Jacobian of the warp for the shape parameters only requires evaluating eqs.2.43 and multiplying it by the precomputed components $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{y}_k}$ as presented in eq.2.41. The projection matrix, $\mathbf{M}_0$, is constant and can be precomputed since a calibrated camera was assumed.

2.4.2 Jacobian of The Warp for The Pose Parameters

The same approach is taken to evaluate the Jacobian of the warp for the pose parameters, that is given by

$$\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}} = \sum_{k=1}^{v} \left[\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{x}_k} \frac{\partial \mathbf{x}_k}{\partial \mathbf{q}} + \frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{y}_k} \frac{\partial \mathbf{y}_k}{\partial \mathbf{q}} \right]. \tag{2.44}$$

A chain rule decomposition is used and the new terms $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j}$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j}$, again both scalars, are found by combining eq.2.8 with eq.2.1, leading to

$$\begin{bmatrix} w\mathbf{x}_k \\ w\mathbf{y}_k \\ w \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{x_k} + \sum_{i=1}^{n} p_i \phi_i^{x_k} + q_j \psi_j^{x_k} + \sum_{i \neq j} q_i \psi_i^{x_k} + s_\psi^{x_k} \\ s_0^{y_k} + \sum_{i=1}^{n} p_i \phi_i^{y_k} + q_j \psi_j^{y_k} + \sum_{i \neq j} q_i \psi_i^{y_k} + s_\psi^{y_k} \\ s_0^{z_k} + \sum_{i=1}^{n} p_i \phi_i^{z_k} + q_j \psi_j^{z_k} + \sum_{i \neq j} q_i \psi_i^{z_k} + s_\psi^{z_k} \\ 1 \end{bmatrix}. \tag{2.45}$$

In the same way, $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j} = \frac{\partial}{\partial \mathbf{q}_j}(\frac{w\mathbf{x}_k}{w})$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j} = \frac{\partial}{\partial \mathbf{q}_j}(\frac{w\mathbf{y}_k}{w})$, resulting in

$$\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j} = \frac{\xi_4 \Xi_6 - \Xi_4 \xi_6}{(\Xi_6)^2} \quad \text{and} \quad \frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j} = \frac{\xi_5 \Xi_6 - \Xi_5 \xi_6}{(\Xi_6)^2} \tag{2.46}$$

with $j = 1, \ldots, 6$ and $k = 1, \ldots, v$. The scalar terms $\xi_4, \xi_5, \xi_6, \Xi_4, \Xi_5, \Xi_6$ are also defined in A.1. Just like in section 2.4.1, the terms $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j}$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j}$ depend both on $\mathbf{p}$ and $\mathbf{q}$ by means of Ξ_4, Ξ_5 and Ξ_6.

Summarizing, both the Jacobians of the warp depend on the current shape and pose parameters, so they are required to be recomputed at every iteration. However, both common components $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{y}_k}$ depend only on the configuration of the projected base mesh, $s_{0_\mathbf{p}}$, and thus can be precomputed and efficiently stored as sparse matrices, reducing the overall computation. At the fitting stage only the computation of $\frac{\partial \mathbf{x}_k}{\partial \mathbf{p}_i}$, $\frac{\partial \mathbf{y}_k}{\partial \mathbf{p}_i}$, $\frac{\partial \mathbf{x}_k}{\partial \mathbf{q}_j}$ and $\frac{\partial \mathbf{y}_k}{\partial \mathbf{q}_j}$, is required, being all scalar values.

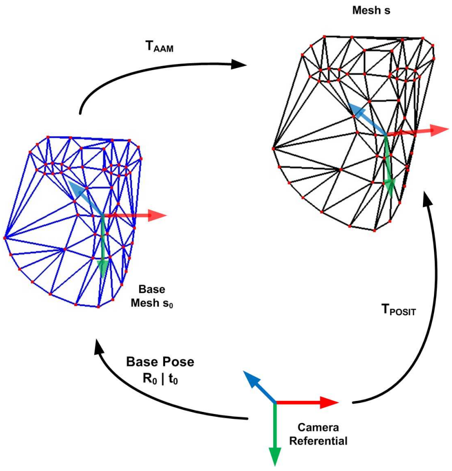


Figure 2.9: The figure shows the coordinate frames involved in the 2.5D AAM. The base pose $\mathbf{T}_0$ is the transformation between the camera and the base mesh s_0, the $\mathbf{T}_{POSIT}$ is the transformation that results from applying POSIT algorithm and $\mathbf{T}_{AAM}$ is the initial transformation required to startup the fitting algorithm. Note that the AAM fitting solves the pose parameters w.r.t. the updated base pose referential and not to the camera.

2.5 The Initial Estimate

The 2.5D AAM requires a rough head pose estimation to establish the initial 3D pose parameters $\mathbf{q}$. From the monocular point of view, estimate the head pose consists on recovering the camera position and relative orientation to a known set of 3D points. In this work the 6DOF pose parameters are estimated using a combination of Adaboost [75] face detection with the Pose from Orthography and Scaling with ITerations (POSIT) [26]. The POSIT algorithm estimates the 6DOF given a set of 3D points (a rigid model) and corresponding 2D image projections. The base mesh s_0 is used as the required 3D rigid model and the 2D correspondences are given by the base mesh projection $s_{0\mathbf{p}}$, scale adjusted to the average AdaBoost detection.

Figure 2.9 shows the different coordinate frames involved in the 2.5D AAM. The camera, the current head position and the base pose referential are shown. The base pose reference, $\mathbf{R}_0, \mathbf{t}_0$, in homogeneous coordinates and represented as $\mathbf{T}_0$, is established during the AAM building process where the training shapes are all aligned into s_0.

The pose estimated by the combination of face detection and POSIT is represented as $\mathbf{T}_{POSIT}$. The initial pose $\mathbf{T}_{AAM}$ is the rigid transformation between the base pose reference and the current head position. Note that the AAM fitting solves the pose parameters w.r.t. the updated base pose referential $(s_0 + s_\psi)$ and not to the camera (applying a further base pose transformation is required to get the 3D mesh points w.r.t the camera frame).

As shown in figure 2.9 the reference frames follow the relationship, $\mathbf{T}_0\mathbf{T}_{AAM} = \mathbf{T}_{POSIT}$, that solving for $\mathbf{T}_{AAM}$, gives

$$\mathbf{T}_{AAM} = \mathbf{T}_0^{-1}\mathbf{T}_{POSIT}. \tag{2.47}$$

2.6 Experimental Results

The 3D shape model (PDM) can be acquired by several ways such as using laser range scans, time-of-flight cameras (ToF), Structure from Motion (SfM) techniques and of course multi-camera networks. The 3D PDM in this work was built using a fully calibrated stereo system where the 2D shape on each view was extracted by fitting a 2D AAM [38] with $v = 58$ landmarks (see supplementary material section A.5 for details). For evaluation purposes a 2.5D AAM was constructed from a set of 20 individuals collected from our institution. A total of 20 images for each individual (10 left + 10 right) exhibiting several expressions and head poses were used in both shape and texture model building process, as described in section 2.2. The 2.5D AAM held $n = 12$ shape parameters, $m = 79$ eigenfaces and the projected base mesh has 68970 gray level pixels (i.e. the figure 2.4-a has size 285×242 pixels).

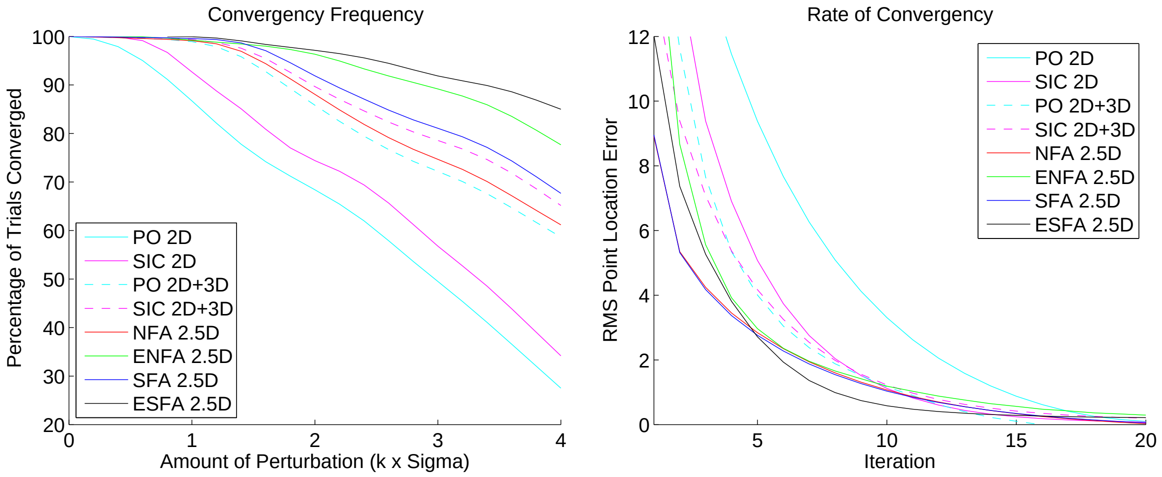
This evaluation compares the projective 2.5D AAM (NFA, SFA, ENFA, and ESFA algorithms) against the state-of-the-art 2D AAM algorithms (Project Out - PO [38] and Simultaneous Inverse Compositional - SIC [79]) and the combined 2D+3D AAM [37] (2D+3D Project Out and 2D+3D Simultaneous Inverse Compositional). Briefly, the 2D+3D AAM [37] uses two shape models: a 2D PDM and a 3D affine PDM built from Non-Rigid Structure-from-Motion (NRSfM). The optimization goal has two main parts (see eq.39 from [37]): the first part deals with pixel intensity matching by optimizing a standard 2D AAM (shape, similarity and appearance parameters) while the second part is a (heavily weighted) soft constrain that enforces the matching between a 3D PDM (scaled orthographic) projection and the current 2D model instance. This constraint ensures that the 2D model deforms according to a valid 3D face projection. The main differences between the 2.5D model and the 2D+3D AAM are: **(1)** The camera projection models. The 2.5D AAM uses a full perspective projection (allowing to retrieve Euclidean metric 3D shapes) whereas the 2D+3D AAM uses a scaled orthographic projection model. **(2)** The model is less complex, using just a single PDM instead of two, and it does not require the NRSfM techniques. **(3)** Finally, the 2D+3D AAM requires to manually tune the weight parameter K that balances the two main terms.

To effectively compare the 2.5D AAM with the 2D+3D AAM an additional 3D (affine) PDM, built from NRSfM, is required. The NRSfM data consists in short video sequences (around 200 frames) taken from the same 20 individuals exhibiting several facial expressions and pose changes. The 2D SIC algorithm was used to fit all the sequences (around 4000 frames in total). Several well known NRSfM algorithms were tested, namely the Xiao-Kanade's method [42][4] (the same approach used in 2D+3D AAM), the Torresani et al. technique that models the shape by a Linear Dynamical System [49] and the NRSfM that uses Discrete Cosine Transform (DCT) basis [36]. In the 2D+3D AAM experiments it was used the state-of-art NRSfM-DCT technique [36] as it proves to be the most reliable in the conducted experiments. Note that, extra Procrustes alignment and PCA are required since no standard basis are given.

2.6.1 Fitting Robustness and Rate of Convergence

To evaluate the fitting robustness and the rate of convergence of the proposed solutions, the performance evaluation scheme presented in [38][37] was adopted. Figure 2.10 shows the results obtained by comparing the fitting robustness and rate of convergence of all the non-robust 2.5D algorithms (NFA, SFA, ENFA, ESFA), the 2D+3D algorithms (PO 2D+3D, SIC 2D+3D) and the standard 2D algorithms (PO 2D, SIC

[4] Code provided by Vincent's Structure from Motion Toolbox [78]



(a) Convergence Frequency (b) Rate of Convergence

Figure 2.10: Fitting and convergence robustness evaluation between the 2.5D, 2D+3D [37] and 2D algorithms [38]. Best viewed in color.

2D).

These experiments measure the performance of the algorithms in two ways: (1) the average frequency of convergence i.e. the number of times each algorithm has converged vs. initial perturbation; (2) the average rate of convergence i.e. the 2D Root Mean Square (RMS) error in the mesh point location vs. iteration number (if convergence was accomplished). For these experiments, each AAM was perturbed from a set of ground truth parameters using independent Gaussian distributions with variance equal to a multiple of a given eigenvalue mode, and tested for convergence. Formally, the parameters disturbance at each experiment was given by

$$\mathbf{p} = \mathbf{p}_{GT} + \mathcal{N}(\mathbf{0}, k\sigma_{\mathbf{p}}) \tag{2.48}$$

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}_{GT} + \mathcal{N}(\mathbf{0}, k\sigma_{\boldsymbol{\lambda}}) \tag{2.49}$$

with an increasing factor $k =]0, 0.1, 0.2, \ldots, 3.9, 4]$. The $\sigma_{\mathbf{p}}$ and $\sigma_{\boldsymbol{\lambda}}$ are the standard deviations from the shape and appearance parameters, respectively. The variances $\sigma_{\mathbf{p}}^2$ and $\sigma_{\boldsymbol{\lambda}}^2$ were estimated at the model building process when applying PCA to both shape and texture models.

The ground truth data was generated using the same AAM by a combination of tracking (say fitting in every frame) / manual initialization / visual confirmation on several small sequences taken from each individual. A subset of 20 random selected frames, from each sequence, was used for further testing, accounting a total of 400 frames. For each testing frame a set of 20 trials was generated by perturbing the shape and appearance parameters simultaneously from the ground-truth (20 trials × 40 noise increasing perturbations experiments per test image). All the algorithms were executed and their convergence ability was evaluated by comparing the final 2D RMS error shape with the ground-truth. A threshold of 1.0 RMS pixels was used to define convergence.

Analyzing figure 2.10, it can be concluded that both 2.5D and 2D+3D fitting algorithms are more robust than 2D algorithms (PO and SIC) and they converge faster, taking fewer iterations to converge. The 3D PDM is inherently higher dimensional than the 2D PDM, however, it uses less 3D shape parameters than the 2D PDM to represent the same visual phenomenon (our PDM has only 12 shape parameters). The 3D PDM is also less prone to local minima because a 2D model can easily generate physically unfeasible shapes, i.e. spanning the 2D PDM parameters can produce a shape that is not even possible, as described in [42][37]. Figure 2.10 also shows that our projective 2.5D AAM performs better than the 2D+3D versions [5].

Besides the full perspective model addition, the 2.5D model outperforms the 2D+3D versions, as it has the following advantages: **(1)** The 2.5D AAM is less dimensional, so less prone to local minima, e.g. the NFA solves $(n+6)$ parameters whereas the PO 2D+3D solves $(n_{2D}+4+n+6)$, namely the 2D shape parameters (n_{2D}), the 4 similarity parameters, the 3D shape parameters (n) and the 6 scaled orthographic camera parameters (the scale, 3D rotations and the 2D translations). **(2)** The optimization uses more accurate gradients. The forwards additive approaches when compared with the inverse compositional (in particular the 2D model from the 2D+3D AAM) produce less

[5] Notice that the methods PO 2D+3D, NFA, ENFA (normalization versions) and SIC 2D+3D, SFA, ESFA (simultaneous versions) should be compared among themselves due to its optimization strategies similarities, i.e. to project out the appearance variation optimizing only the shape and pose or optimizing all parameters at once, respectively.

second order terms in the Taylor series approximation (eq.2.13). The main optimization neglects more terms if an inverse compositional method is used [64]. This means that our forwards additive 2.5D use gradients that are closer to the "true" gradients, being therefore more accurate and take less iterations to converge (as shown by figure 2.10-b). **(3)** As previously mentioned, the 2D+3D AAM requires tuning a constant K that weights the 3D affine projection constraint. When K is too small (soft constant) the combined model fits a 2D and a 3D shape independently (the 3D projection and the 2D model do not converge). However, if K is set to be too large, e.g. 10^6, the gradient descent updates (times the inverse of the Hessian) are too small, and the model requires a lot more iterations to converge. In all experiments K was set to $K = 10000$. The 2.5D model does not have this weighting issue. **(4)** The 2D+3D AAM requires to compute Jacobians for the constraints w.r.t. the 2D shape and pose parameters ($\frac{\partial Ft_i}{\partial \mathbf{p}} \mathbf{J_p}$ and $\frac{\partial Ft_i}{\partial \mathbf{q}} \mathbf{J_q}$), which are not required for the 2.5D model. Furthermore, these Jacobians are numerically estimated. **(5)** A minor, but still an advantage, is that our model do not require to inverse compose the warp at each iteration since the parameters update is additive. Both 2D versions and consequently the 2D+3D versions require to inverse compose the warp at each iteration, which still is an approximation (averaging the neighbor triangles) since no true inverse exists [38]. **(6)** Finally, the 2.5D AAM is a lot more simple and easier to implement when compared to the 2D+3D model.

The results also show that the efficient versions (ENFA, ESFA) perform even better than the standard formulations. The main reason for this performance increase is the reduced noise influence that comes out from avoiding the reevaluation of the gradients of the input image in each iteration, as described in section 2.3.3. The Efficient-SFA, that searches simultaneously for all the parameters, has proved to be the best algorithm w.r.t. convergence speed showing high fitting success rates even from far initial estimates.

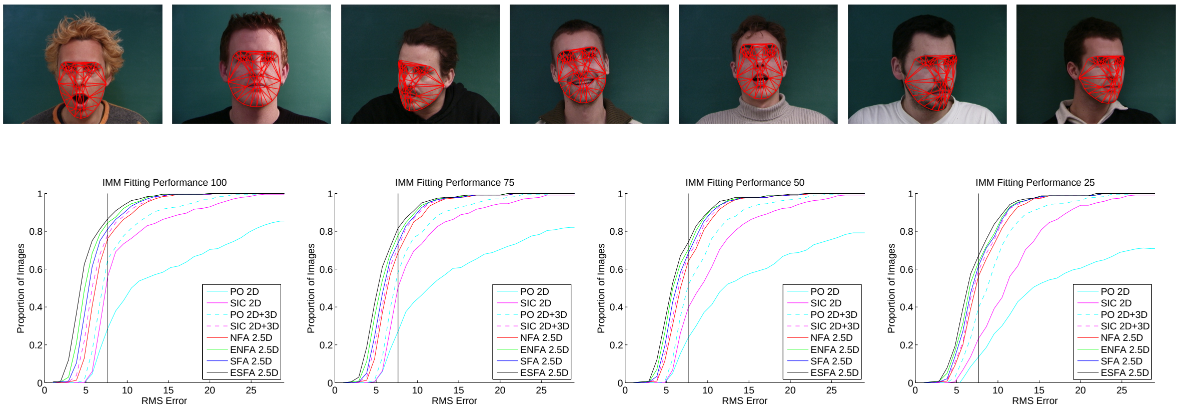
2.6.2 Performance in Unseen Data

The AAM is a generative (holistic) method as it models the appearance of all image pixels within the face. By synthesizing the expected appearance template it achieves a high registration accuracy on the dataset it was trained for but it performs poorly in unseen data (individuals not captured by the texture PCA). If the appearance of a target individual does not lie in the subspace spanned by $\mathbf{A}_i(\mathbf{x_p})$, the AAM can not generate a good template and the model fitting will not converge.

The AAM fitting performance in unseen data was evaluated by running a series of experiments, changing the amount of training images in the model building process. The IMM [57] database was used, as it consists of 240 annotated images (58 ground truth landmarks) of 40 different human faces presenting different head pose, illumination and facial expression.

All the fitting algorithms that previously appeared in section 2.6.1 were used in this evaluation, namely PO 2D, SIC 2D, PO 2D+3D, SIC 2D+3D, NFA, SFA, ENFA and SFA. Four main experiments were conducted, training all the algorithms with the full sized dataset (100% - 240 images), then 75% (180 images), 50% (120 images) and finally only 25% (60 images) of the dataset. Then, all the runs were evaluated by fitting the entire IMM set.

Figure 2.11 shows the fitting performance curves for these four experiments. These are standard curves that show the percentage of faces that converge with less or equal Root Mean Square (RMS) error amount. The table in the same figure shows quantitative values taken by sampling the graphics using a fixed RMS error amount (7.5 pixels - represented as the vertical line in the graphics). As expected, all the methods reveals a fitting performance decrease (less images converge for the same RMS value) as the appearance representation power decrease. The relative performance between all the methods are conform to section 2.6.1. The 2D models have the lower performance (where the Project Out performs the worst), followed by the combined 2D+3D model and then our projective 2.5D versions, where the efficient algorithms perform the better. According, the simultaneous versions perform better than the error normalization versions mainly due to their improved search strategy (all parameters at once). The overall results show that using a 3D PDM projection effectively increases the performance in unseen data.



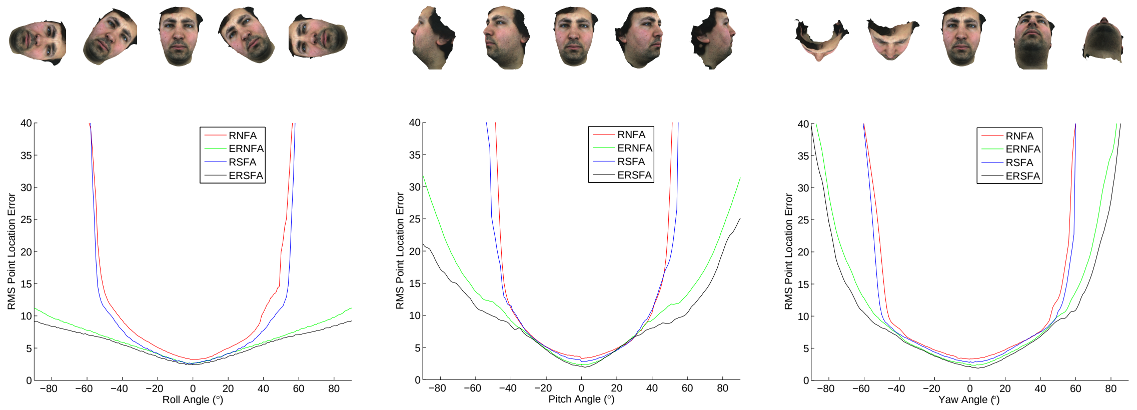
(a) 100% (b) 75% (c) 50% (d) 25%

Reference 7.5 RMS	PO 2D	SIC 2D	PO 2D+3D	SIC 2D+3D	NFA 2.5D	ENFA 2.5D	SFA 2.5D	ESFA 2.5D
(a) 100%	28.7	56.2	65.4	78.3	76.2	84.6	81.2	**86.7**
(b) 75%	27.1	50.4	58.3	71.2	68.8	77.9	73.8	**81.7**
(c) 50%	23.8	39.6	51.7	67.1	64.2	70.4	68.3	**74.2**
(d) 25%	13.6	23.3	39.6	59.2	56.2	63.7	61.7	**67.1**

Figure 2.11: Fitting performances curves on the IMM [57] database using 100%, 75%, 50% and 25% of training images, respectively. The table shows quantitative values taken by sampling the graphics using a fixed RMS error amount (7.5 pixels - represented as the vertical line). Each table entry show how many percentage of images converge with less or equal RMS error that the reference. Top images show fitting examples from the IMM database using the ESFA algorithm.

2.6.3 Robust Methods Evaluation

The robust fitting methods proposed in this work intend to improve the performance w.r.t. self occlusion due to 3D head motion. To evaluate these algorithms, namely the RNFA, the RSFA and the efficient versions ERNFA and ERNFA, three synthetic sequences were created. A set of images with an individual standing in near frontal position was used. The current 3D mesh location was found by fitting the 2.5D AAM using ESFA. Then, ranging the 3D mesh from $-90°$ to $90°$ degrees in both roll, pitch and yaw angles, using one degree of resolution, the fixed appearance image was projected into the camera and stored (figure 2.12-top). Finally, all the robust fitting algorithms were evaluated using these sequences, starting from the frontal position. In all the algorithms the scale parameters, $\sigma_{\mathbf{x_p}}$, were estimated from the fitting error MAD.



(a) Roll (b) Pitch (c) Yaw

Figure 2.12: Robust algorithms evaluation on synthetic sequences at top figure. The graphics show the RMS error due to roll, pitch and yaw angles ranging from −90° to 90°, respectively.

Figure 2.12-bottom shows the RMS error in point location for all the algorithms. Once again the Efficient versions of the algorithms (ERNFA and ERSFA) outperform their standard versions (RNFA and RSFA). Also, the ERSFA performs slightly better that the ERNFA, as expected, due to the parameters search strategy. These experiments show that, using the efficient algorithms, the model can successfully deal with rotations of almost ±90° in roll, pitch and yaw angles, respectively.

2.6.4 Results on the BU-4DFE Dataset

This section evaluates the quality of the 3D recovered shape when using the 2.5D AAM. The Binghamton University 3D Dynamic Facial Expression Database (BU-4DFE) [50] was used for this evaluation process. The BU-4DFE dataset includes high resolution 3D dense reconstructions of video sequences of several individuals showing the six prototypic facial expressions [21] namely, anger, disgust, happiness, fear, sadness, and surprise. The 3D facial expressions were captured at 25 frames per second where each expression sequence contains about 100 frames (resolution of 1040 × 1392 per frame).

Due to the generative nature of the AAM, a new BU-4DFE tuned model must be built to run these experiments. To fit every frame of the database the AAM should hold as much shape variation as possible. To accomplished this the training images were composed by the most emotion expressive images of the testing set. These training images were hand annotated using also the 58 landmarks scheme ($v = 58$). Holding 95% of the shape and appearance variance produces a 2.5D AAM with 19 shape parameters, ($n = 19$), and 87 eigenfaces, ($m = 87$). The projected base mesh width was set to 300 pixels, as described in the 3D model building process in supplementary material, resulting on a total of 100500 gray level pixels used by the appearance model.

In this section, only the ESFA algorithm has been used, because it was shown previously to be the most accurate. A subset of the BU-4DFE dataset, consisting in 7 males and 7 females, forming a total of around 8400 frames were used in this evaluation. The ESFA algorithm was applied on every frame of each sequence for all the testing subjects, and the RMS error between the current PDM shape s (the shape that the model fits for) and the ground truth extracted from the BU-4DFE dataset was evaluated.

The shape RMS error is given by

$$e_{\mathrm{RMS}}(s) = \sqrt{\frac{1}{v}\sum_{i=1}^{v}\left(s^{x_i} - s_{gt}^{x_i}\right)^2 + \left(s^{y_i} - s_{gt}^{y_i}\right)^2 + \left(s^{z_i} - s_{gt}^{z_i}\right)^2} \tag{2.50}$$

where the ground truth shape, s_{gt}, was extracted from the dense reconstruction by lookup the 3D depth from the 2D image projections found by the 2.5D AAM.

Figure 2.13 shows examples of the AAM fitting, the correspondent 3D dense reconstruction ground truth and a graphic showing the RMS shape error over time for each emotion sequence (for a single test subject). The evaluation shows that globally, during the entire sequence, the fitting error stays low, exhibiting an average error of around 5mm (in 3D space). Typically, in the captured facial expression sequences of the BU-4DFE dataset, each individual starts from a neutral expression, exhibits the emotion until its maximum intensity and then goes back to the neutral state. The graphic shows that the RMS error match this behavior, i.e. the AAM has a lower shape fitting error during the begin and at end of the sequences when the individual displays the neutral emotion. The results also show that the surprise facial expression

	Angry	Disgust	Fear	Happy	Sad	Surprise	**Overall**
avg	4.51	4.82	4.92	4.78	4.64	6.28	4.99
std	0.42	0.83	0.92	0.36	0.82	1.67	0.83

Table 2.1: The RMS shape fitting error over a subset of the BU-4DFE dataset consisting of 14 individuals (7 males and 7 females). About 8400 frames were used. The table show the mean and standard deviation found for each facial expression sequence and also for the entire set (overall). The units are in mm.

is the one that holds more fitting error, mainly because it is the emotion that more deforms the face from the neutral state.

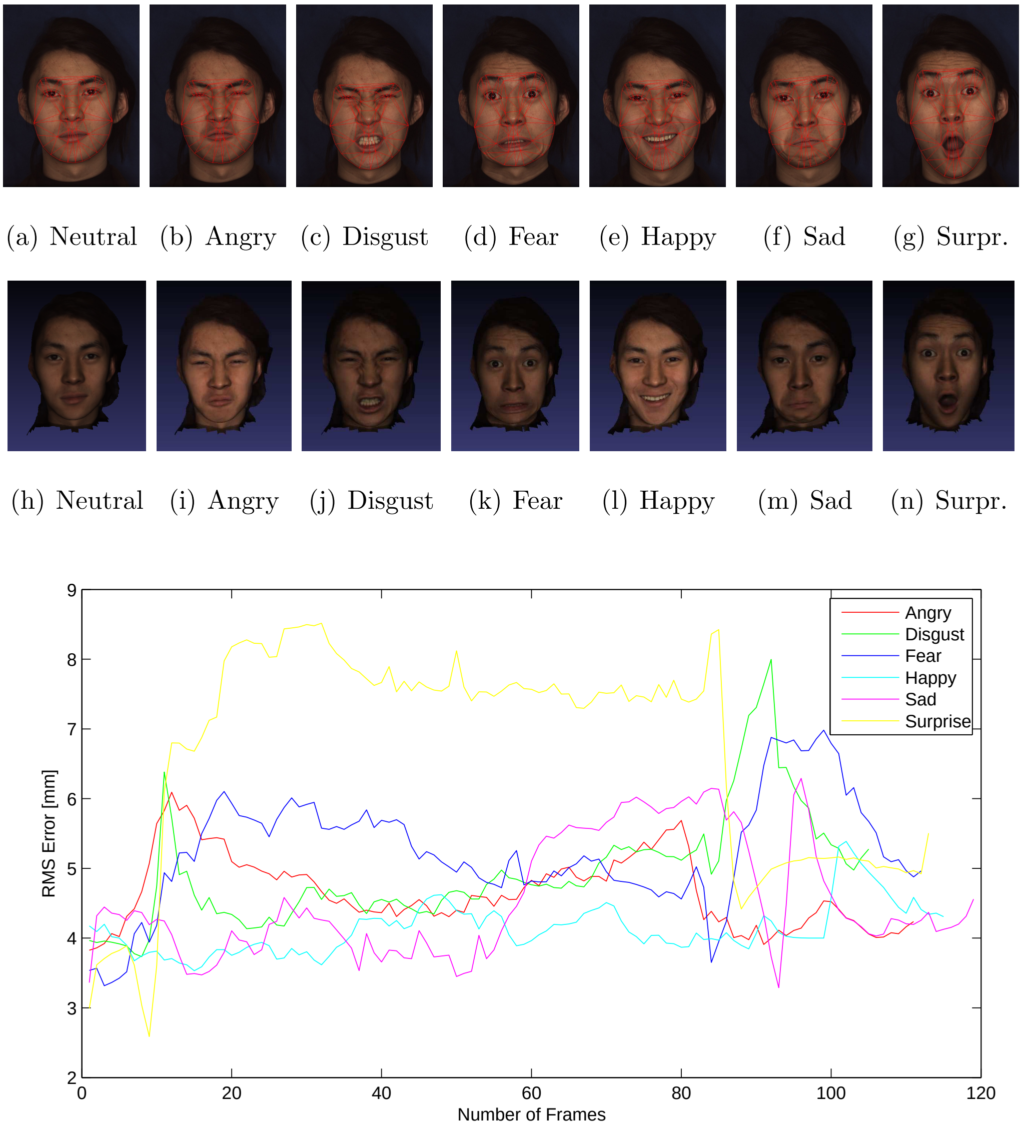
Table 2.1 displays the mean and standard deviations of the RMS shape error over the entire testing subset of the BU-4DFE.

2.6.5 Tracking Performance

The tracking performance is evaluated on the challenging FGNet Talking Face (TF) [32] video sequence that holds 5000 frames of video of an individual engaged in a conversation. The full sequence is annotated using 68 landmarks (2D ground truth). Just like in previous sections (2.6.1 and 2.6.2), all AAM algorithms are used, namely the PO 2D, SIC 2D, PO 2D+3D, SIC 2D+3D, NFA, SFA, ENFA and SFA. A minor difference from the previous experiments is that a few annotated frames from the TF sequence were added in each AAM so that the appearance model $\mathbf{A}_i(\mathbf{x_p})$ can now include the new individual.

The figure 2.14 shows the RMS fitting error for all the evaluated methods. Since we are using a 58 landmark scheme and the TF uses 68, the error was only measured over the correspondent landmarks. The quantitative values on the legend box are the mean and standard deviation values for the RMS error.

Globally, as expected, all the 2.5D algorithms (NFA, ENFA, SFA and ESFA) perform better than the 2D algorithms (especially when exists some degree of head pose variation) and slightly better than the 2D+3D algorithm, confirming their relative performance. Again, the efficient versions also express a performance advantage over all



(a) Neutral (b) Angry (c) Disgust (d) Fear (e) Happy (f) Sad (g) Surpr.

(h) Neutral (i) Angry (j) Disgust (k) Fear (l) Happy (m) Sad (n) Surpr.

Figure 2.13: Evaluation of the 3D recovered shape when using the 2.5D AAM. The top a)-g) figures show examples of AAM fitting on a test subject of the BU-4DFE [50] database exhibiting the six basic emotions plus the neutral one. Images h)-n) shows the correspondent 3D dense reconstruction provided by the database. The ground truth, s_{gt}, used in all evaluations is a sparse shape that results from retrieving the 3D data from the dense reconstruction on the 2D projections points found by the AAM (the red mesh at top figures). The bottom graphic show the RMS shape error during each of the facial expressions sequences of the testing individual shown in the top images. The RMS error units are in mm. A 2.5D AAM fiting video showing some examples of BU-4DFE dataset can be seen at http://www.isr.uc.pt/~pedromartins/Videos/PhD.

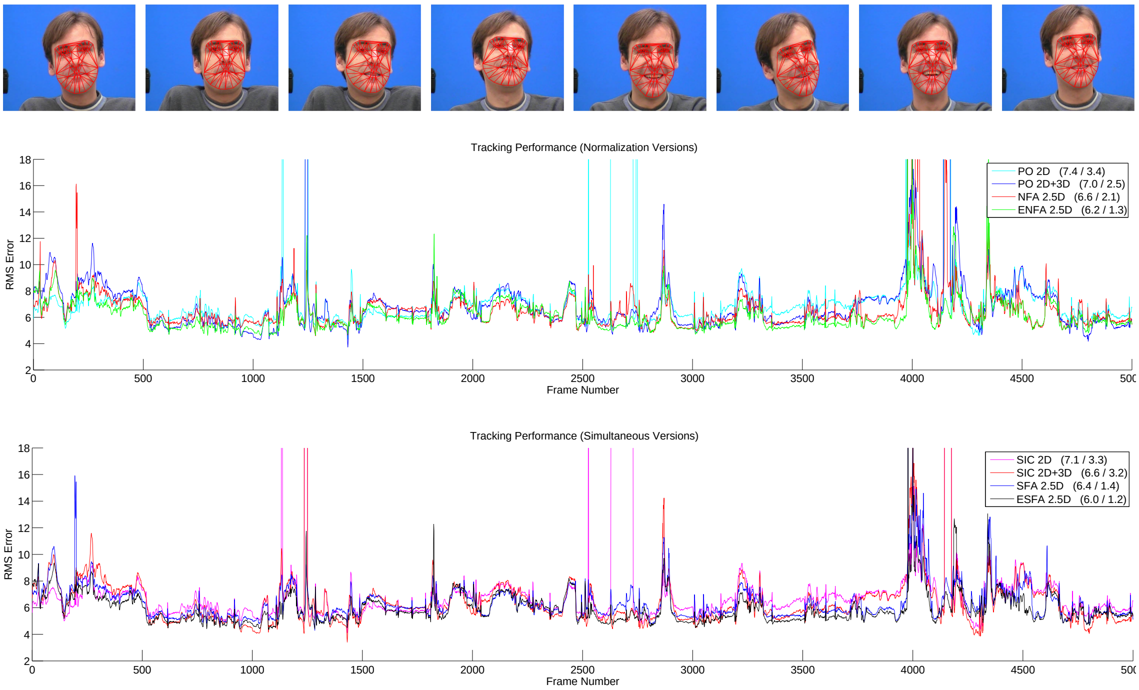


Figure 2.14: RMS shape error on the Talking Face [32] video sequence. The top images show ESFA fitting examples. The values on legend box are the mean and standard deviation RMS errors, respectively. Best viewed in color.

the others.

2.6.6 Head Pose Estimation

As described previously, one of the main advantage of using the 2.5D AAM is its ability to recover the 3D Euclidean shapes from a single image. The figure 2.15-top shows some of these examples taken from the TF video sequence using the ERSFA algorithm. Only qualitative results are shown because the TF dataset does not provide 3D ground truth information.

Although the proposed methods are not explicitly oriented for pose estimation, the updates on the pose parameters, $\Delta\mathbf{q}$, can be analyzed and used for this purpose. The pose is, therefore, estimated between the camera coordinate frame and the rigid component of the PDM (given by $s_0 + s_\phi$) at each frame. The bottom part of figure 2.15 show the pose estimation results taken from the TF sequence. Again, only qualitative

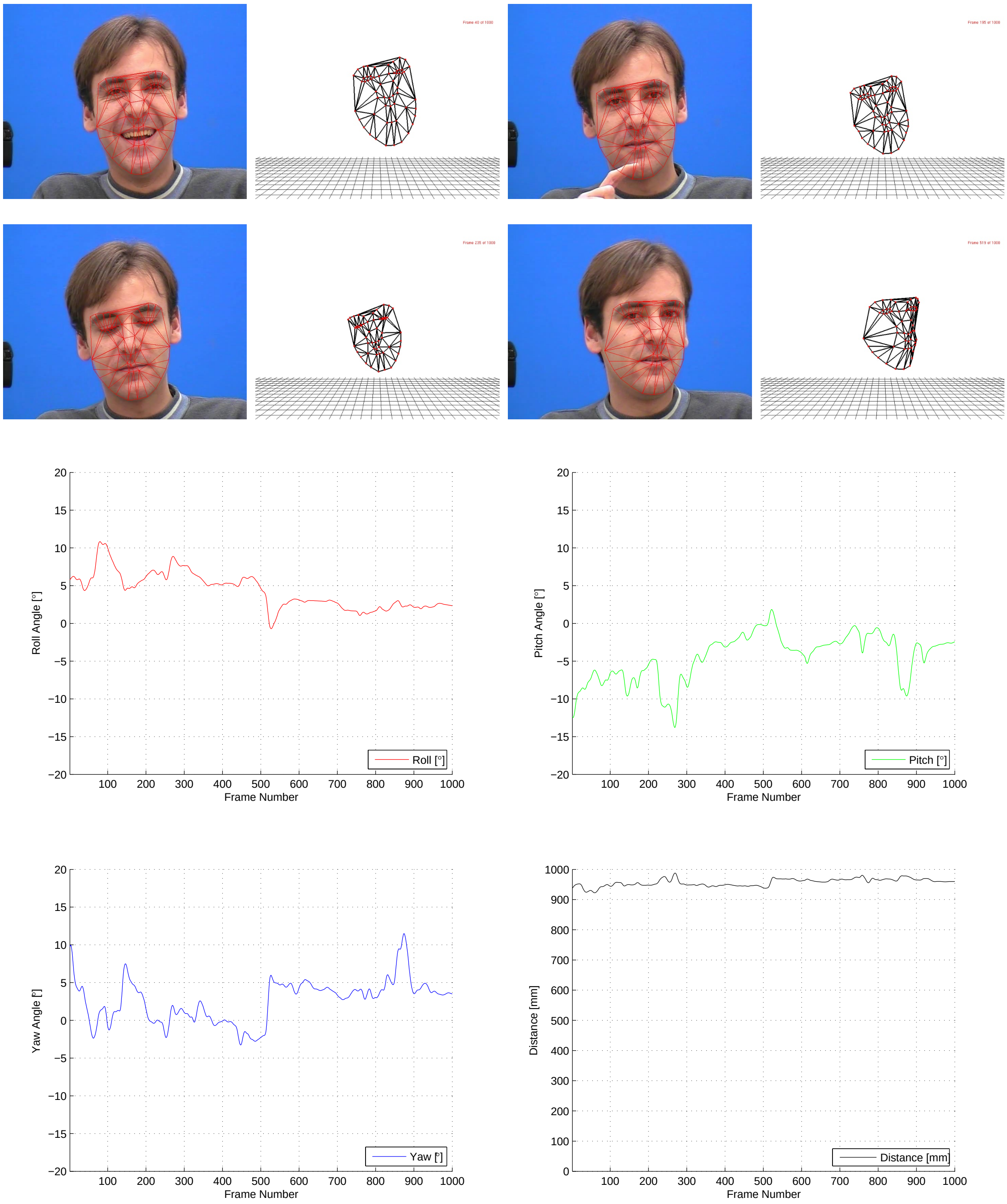


Figure 2.15: The top images show qualitative 3D shape recovery results on the first 1000 frames of the Talking Face [32] video sequence. The graphics show the estimated roll, pitch and yaw angles (in degrees) and distance (in mm) to camera. The ERSFA algorithm was used in both experiments. The full video sequence can be seen at `http://www.isr.uc.pt/~pedromartins/Videos/PhD`.

	Normalization Versions				Simultaneous Versions			
	PO 2D	PO 2D+3D	NFA 2.5D	ENFA 2.5D	SIC 2D	SIC 2D+3D	SFA 2.5D	ESFA 2.5D
Time per iteration	310	340	1780	760	460	550	1820	780

Table 2.2: Fitting times by iteration on the evaluated algorithms. The present times are in ms taken using a MatLab implementation. Note that the 2.5D methods even being slower, they require less iterations to converge.

results are shown due to the lack of 3D ground truth.

2.6.7 Computational Performance

Table 2.2 shows a comparison, in computational cost, between all the evaluated algorithms during the entire section 2.6. The table shows approximated fitting times per iteration using a MatLab implementation on a 3GHz Intel i7 CPU with 4GB of RAM running Fedora 14 OS. All the AAM use the same settings mentioned on section 2.6.1.

The 2D Projected Out is probably the fastest approach introduced so far, where both the Jacobian and the Hessian matrices are constant and can be precomputed. The 2D + 3D PO only requires to reevaluate the Jacobians for the constraints and parts of the Hessian (most part is constant). The simultaneous extensions (SIC 2D and SIC 2D+3D) are much slower because they must evaluate the SD images, the Hessian and its inverse on a larger set of parameters that now include the appearance parameters. As shown in algorithms 1, 2, 10 and 11 the 2.5D algorithms need to perform image warping (it takes around 1200ms and 220ms in the standard and efficient versions, respectively), recompute the SD images (around 400ms) and the Hessian. Even being slower, they require less iterations to converge as shown in figure 2.10-b. However a C/C++ version of ESFA achieves near real-time performance (around 10 fps - using a base mesh with almost 70K pixels). Additional speed up can be achieved by reducing the base mesh size.

2.7 Conclusions

In this chapter we presented a novel formulation for 3D facial image alignment from single view 2D images through a 2.5D AAM. The major contribution of the chapter lies on the use of a 2.5D AAM that combines a 3D metric PDM with a full perspective projection model that defines the 2D appearance. The 2.5D AAM is able to recover 3D Euclidean shapes by assuming a calibrated camera. Two algorithms and computational efficient approximations are proposed, both based on the Lucas and Kanade framework: the Simultaneous Forwards Additive (SFA) and the Normalization Forwards Additive (NFA). The SFA, when compared with NFA, is the most accurate algorithm and also the most computationally expensive. Their efficient versions have shown a substantial improvement in the fitting performance, being more robust to noise and able to converge from far initial estimates, requiring less computational effort. To make the model able to deal with self or partial occlusion, robust extensions to SFA and NFA were also proposed. Again, their efficient approximations perform much better that the basic versions. Several performance evaluations carried out on real an synthetic data demonstrated that the 2.5D AAM algorithms outperform both the combined 2D+3D AAM and the traditional 2D AAM algorithms and accurately handle face pose variations. Finally, the quality of the 3D retrieved shape was also evaluated. The performed tests on the BU-4DFE [50] database show that the 2.5D AAM is an effective method to recover the 3D Euclidean shape.

Chapter 3

Discriminative Bayesian Active Shape Models

This chapter presents a simple and very efficient solution to align facial parts in unseen images. The proposed approach is closely related to Constrained Local Models (CLM) and Active Shape Models (ASM), where an ensemble of local feature detectors are constrained to lie within the subspace spanned by a Point Distribution Model (PDM). Fitting a model to an image typically involves two steps: a local search using a detector, obtaining response maps for each landmark (likelihood term) and a global optimization that finds the PDM parameters that jointly maximize all the detection responses. The global optimization can be seen as a Bayesian inference problem, where the posterior distribution of the PDM parameters (including pose) can be inferred in a *maximum a posteriori* (MAP) sense. However, previous formulations do not model explicitly the covariance of the latent variables, which represents the confidence in the current solution. In the Discriminative Bayesian Active Shape Model (DBASM) formulation, described here, the MAP global alignment is inferred by a Linear Dynamical System (LDS) that takes this information into account. The Bayesian paradigm provides an effective fitting strategy, since it combines in the same framework both the shape prior and multiple sets of patch alignment classifiers.

In later work, the previous DBASM formulation was extended to explicitly model the prior distribution. A second global optimization, Bayesian Active Shape Model (BASM) is presented, where the prior term is used to encode the dynamic transitions of the PDM parameters. Using recursive Bayesian estimation, the prior distribution of the data is modeled as being Gaussian. The mean and covariance were assumed to be unknown and treated as random variables.

Both DBASM and BASM extensive evaluations were performed on several standard datasets (IMM, BioID, XM2VTS and FGNET Talking Face) against state-of-the-art methods while using the same local detectors. Face parts descriptors were also evaluated, including the recently proposed Minimum Output Sum of Squared Error (MOSSE) filter. It is demonstrated that generic image alignment by explicitly modelling the prior distribution (BASM) offers a significant increase in performance. Finally, qualitative results taken from the challenging Labeled Faces in the Wild (LFW) dataset are also shown.

Publications

The contents of this chapter resulted in two main publications:

- Discriminative Bayesian Active Shape Models [71]
  Pedro Martins, Rui Caseiro, João F. Henriques, Jorge Batista
  ECCV 2012 - European Conference on Computer Vision

- Let the Shape Speak - Discriminative Face Alignment using Conjugate Priors [72]
  Pedro Martins, Rui Caseiro, Joäo F. Henriques, Jorge Batista
  BMVC 2012 - British Machine Vision Conference [**Oral Presentation**]

3.1 Introduction

Deformable model fitting aims to find the parameters of a Point Distribution Model (PDM) that best describe the object of interest in an image. Several fitting strategies have been proposed, most of which can be categorized as being either holistic (generative) or patch-based (discriminative). The holistic representations [99][38] model the appearance of all image pixels describing the object. By synthesizing the expected appearance template, a high registration accuracy can be achieved. However, such representation generalizes poorly when the object of interest exhibits large amounts of

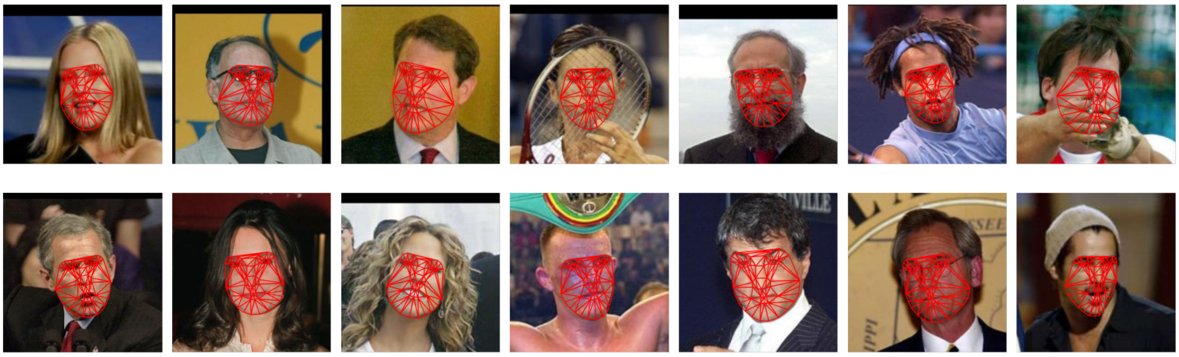


Figure 3.1: Examples of the DBASM global alignment on the LFW [33] dataset. Video at http://www.isr.uc.pt/~pedromartins/Videos/PhD.

variability, such as the case of the human face under variations of identity, expression, pose, lighting or non-rigid motion, due to the huge dimensional representation of the appearance (learnt from limited data).

Recently, discriminative-based methods, such as the Constrained Local Model (CLM) [98][116][24][74][25][115], have been proposed. These approaches can improve the model's representation capacity, as it accounts only for local correlations between pixel values. In this paradigm, both shape and appearance are combined by constraining an ensemble of local feature detectors to lie within the subspace spanned by the PDM. The CLM implements a two step fitting strategy: a local search and a global optimization. The first step performs an exhaustive local search using a feature detector, obtaining response maps for each landmark. Then, the global optimization finds the PDM parameters that jointly maximize the detection responses. Each landmark detector generates a likelihood map by applying local detectors to the neighborhood regions around the current estimate.

Some of the most popular optimization strategies propose to replace the true response maps by simple parametric forms (Weighted Peak Responses [98], Gaussians Responses [115], Mixture of Gaussians [46]) and perform the global optimization over these forms instead of the original response maps. The detectors are learned from training images of each of the object's landmarks. However, due to their small local support and large appearance variation, they can suffer from detection ambiguities. In

[41] the authors attempt to deal with these ambiguities by nonparametrically approximating the response maps using the mean-shift algorithm, constrained to the PDM subspace (Subspace Constrained Mean-Shift - SCMS). However, in the SCMS global optimization the PDM parameters update is essentially a regularized projection of the mean-shift vector for each landmark onto the subspace of plausible shape variations. Since a least squares projection is used, the optimization is very sensitive to outliers (when the mean-shift output is very far away from the correct landmark location). The patch responses can be embedded into a Bayesian inference problem, where the posterior distribution of the global warp can be inferred in a *maximum a posteriori* (MAP) sense. The Bayesian paradigm provides an effective fitting strategy, since it combines in the same framework both the shape prior (the PDM) and multiple sets of patch alignment classifiers to further improve the accuracy.

3.1.1 Main Contributions

1. A novel and efficient Bayesian formulation is presented to solve the MAP global alignment problem (Discriminative Bayesian Active Shape Model - DBASM). The main advantage of the proposed DBASM with respect to the previous Bayesian formulations is that we model the covariance of the latent variables, which represents the confidence in the current parameters estimate i.e. DBASM explicitly maintains 2$^{\text{nd}}$ order statistics of the shape and pose parameters, instead of assuming them to be constant. It is shown that the posterior distribution of the global warp can be efficiently inferred using a Linear Dynamical System (LDS) taking this information into account.

2. It is shown that aligning the PDM using a Bayesian approach offers a significative increase in performance, in both fitting still images and video sequences, when compared with state-of-the-art first order forwards additive methods [98][115][41]. We confirm experimentally that the MAP parameter update outperforms the standard optimization strategies, based on maximum likelihood solutions (least squares). See figures 3.6 and 3.7.

3. A comparison between several face parts descriptors is presented, including the recently proposed Minimum Output Sum of Squared Error (MOSSE) filters [28]. The MOSSE maps aligned training patch examples to a desired output, producing correlation filters that are notably stable. These filters exhibit a high invariance to illumination, due to their null DC component. Results show that the MOSSE outperforms the others detectors, being particularly well-suited to the task of generic face alignment (figures 3.2 and 3.5).

4. A second Bayesian global optimization strategy is presented (Bayesian Active Shape Models - BASM) designed to infer both the PDM and the pose parameters, in a MAP sense, by explicitly modelling the prior distribution (encoding the dynamic transitions of the PDM parameters). Using recursive Bayesian estimation, the prior distribution of the data is modeled as being Gaussian. The mean and covariance were assumed to be unknown and treated as random variables. This means that, not only the mean and the covariance are estimated but also the probability distribution of the mean and the covariance (using conjugate priors).

5. Extensive evaluations were performed on several standard datasets (IMM [57], BioID [61], XM2VTS [45] and FGNET Talking Face [32]) against state-of-the-art methods while using the same local detectors. Qualitative results of the challenging Labeled Faces in the Wild (LFW) [33] dataset are also shown.

3.1.2 Outline

This chapter is organized as follows: **Section** 3.2 briefly explains the shape model PDM. **Section** 3.3 presents our DBASM global optimization approach. Experimental results comparing the fitting performances of several local detectors (including the MOSSE filters) and several global optimizations strategies are shown in **Section** 3.4. The BASM extended global optimization is introduced and described in **Section** 3.5. Performance evaluation experiments, following the previously determined protocol, is shown in **Section** 3.6. Finally, **Section** 3.7 provides the overall conclusions.

3.2 The Shape Model - PDM

The shape $\mathbf{s}$ of a Point Distribution Model (PDM) is represented by the 2D vertex locations of a mesh, with a $2v$ dimensional vector $\mathbf{s} = (x_1, y_1, \ldots, x_v, y_v)^T$. The traditional way of building a PDM requires a set of shape annotated images that are previously aligned in scale, rotation and translation by Procrustes Analysis [15]. Applying a PCA [56] to a set of aligned training examples, the shape can be expressed by the linear parametric model

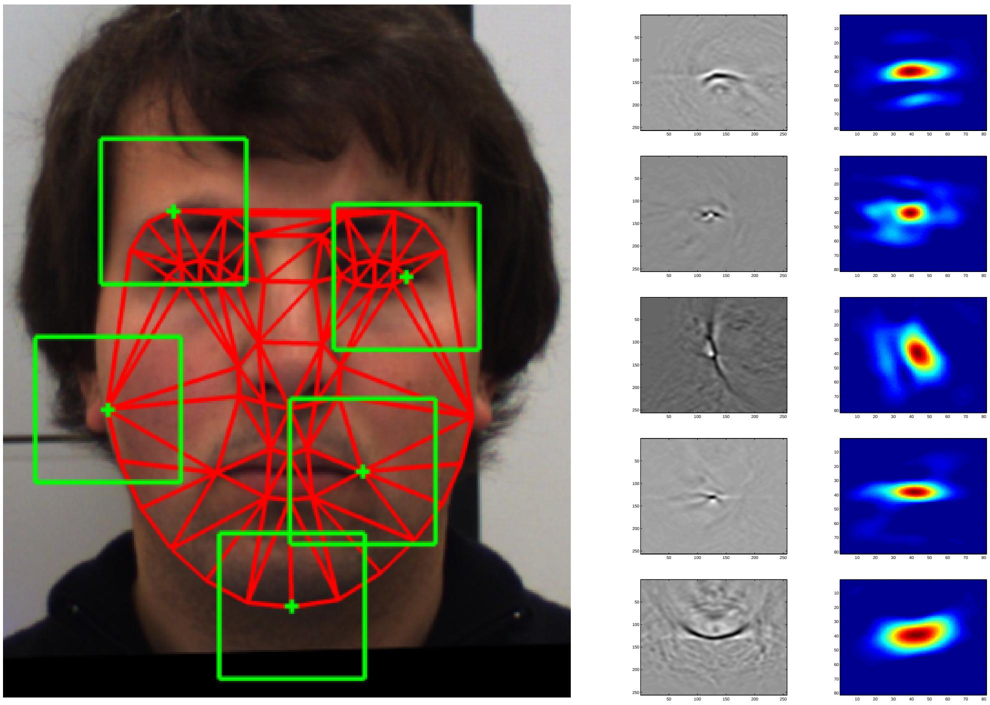
$$\mathbf{s} = \mathcal{S}\left(\mathbf{s}_0 + \Phi \mathbf{b}_s, \mathbf{q}\right) \tag{3.1}$$

where $\mathbf{s}_0 = (x_1^0, y_1^0, \ldots, x_v^0, y_v^0)^T$ is the mean shape (also referred to as the base mesh), Φ is the shape subspace matrix holding n eigenvectors (retaining a user defined variance, e.g. 95%), $\mathbf{b}_s$ is a vector of shape parameters, $\mathcal{S}(., \mathbf{q})$ represents a similarity transformation function of the $\mathbf{q}$ pose parameters. Defining the pose parameters to be $\mathbf{q} = (s\cos(\theta) - 1, s\sin(\theta), t_x, t_y)^T$ (where s, θ, t_x, t_y are the scale, rotation and translations w.r.t. the base mesh $\mathbf{s}_0$, respectively) and Ψ to be a matrix holding four special eigenvectors $\Psi = [\psi_1\ \psi_2\ \psi_3\ \psi_4]$ with $\psi_1 = \mathbf{s}_0$, $\psi_2 = (-y_1^0, x_1^0, \ldots, -y_v^0, x_v^0)^T$, $\psi_3 = (1, 0, \ldots, 1, 0)^T$ and $\psi_4 = (0, 1, \ldots, 0, 1)^T$, the 2D pose can be linearly represented [38] (i.e. the amount $\mathbf{s}_0 + \Psi\mathbf{q}$ represents the same geometric change than applying a generic 2D similarity transformation to $\mathbf{s}_0$).

From the probabilistic point of view, $\mathbf{b}_s$ follows a multivariate Gaussian distribution $\mathbf{b}_s \propto \mathcal{N}(\mathbf{b}_s|\mathbf{0}, \Lambda)$, with $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where λ_i denotes the PCA eigenvalue of the i^{th} mode of deformation.

3.3 Global PDM Optimization - DBASM

This section describes the proposed global optimization method (Discriminative Bayesian Active Shape Models - DBASM). The deformable model fitting goal (that follows the parametric form eq.3.1) is formulated as a global shape alignment problem in a *maximum a posteriori* (MAP) sense.



(a) Example image and search regions. (b) $\mathcal{F}^{-1}\{\mathbf{H}_i^*\}$ (c) $p_i(\mathbf{z}_i)$

Figure 3.2: The DBASM combines a Point Distribution Model (PDM) and a set of discriminant local detectors, one for each landmark. a) Image with the current mesh showing the search region for some landmarks. b) The local detector (the MOSSE filter [28] itself). c) Response maps for the correspondent highlighted landmarks. The DBASM global optimization jointly combines all landmark response maps, in a MAP sense, using 2$^{\text{nd}}$ order statistics of the shape and pose parameters.

3.3.1 The Alignment Goal

Given a $2v$ vector of observed positions $\mathbf{y}$, the goal is to find the optimal set of parameters $\mathbf{b}_s^*$ that maximizes the posterior probability of being its true position. Using a Bayesian approach, the optimal shape parameters are defined as

$$\mathbf{b}_s^* = \arg\max_{\mathbf{b}_s} p(\mathbf{b}_s|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{b}_s)p(\mathbf{b}_s) \tag{3.2}$$

where $\mathbf{y}$ is the observed shape, $p(\mathbf{y}|\mathbf{b}_s)$ is the likelihood term and $p(\mathbf{b}_s)$ is a prior distribution over all possible configurations. The section 3.3.2 describe some possible strategies to set the observed shape vector $\mathbf{y}$.

The complexity of the problem, in eq.3.2, can be reduced by making some simple assumptions. Firstly, conditional independence between landmarks can be assumed simply by sampling each landmark independently. Secondly, it can also be considered that we have an approximate solution to the true parameters ($\mathbf{b} \approx \mathbf{b}^*_s$). Combining these approximations, the eq.3.2 can be rewritten as

$$p(\mathbf{b}|\mathbf{y}) \propto \left(\prod_{i=1}^{v} p(\mathbf{y}_i|\mathbf{b}) \right) p(\mathbf{b}|\mathbf{b}^*_{k-1}) \tag{3.3}$$

where $\mathbf{y}_i$ is the i^{th} landmark coordinates and $\mathbf{b}^*_{k-1}$ is the previous optimal estimate of $\mathbf{b}$.

3.3.2 The Likelihood Term

The likelihood term, including the PDM model (in eq.3.1), becomes the following convex energy function:

$$p(\mathbf{y}|\mathbf{b}) \propto \exp \left(-\frac{1}{2} (\underbrace{\mathbf{y} - (\mathbf{s}_0}_{\Delta\mathbf{y}} + \Phi\mathbf{b}))^T \Sigma_{\mathbf{y}}^{-1} (\mathbf{y} - (\mathbf{s}_0 + \Phi\mathbf{b})) \right) \tag{3.4}$$

where $\Delta\mathbf{y}$ is the difference between the observed and the mean shape and $\Sigma_{\mathbf{y}}$ is the uncertainty of the spatial localization of the landmarks ($2v \times 2v$ block diagonal covariance matrix). From the probabilistic point of view, the likelihood term follows a Gaussian distribution given by

$$p(\mathbf{y}|\mathbf{b}) \propto \mathcal{N}(\Delta\mathbf{y}|\Phi\mathbf{b}, \Sigma_{\mathbf{y}}). \tag{3.5}$$

Finding the Likelihood Parameters - Local Optimization Strategies

This section briefly describes several local strategies to represent the true response maps by a probabilistic model (parametric and nonparametric). We also describe how to extract from each probabilistic model the likelihood term of the MAP formulation (observed shape $\mathbf{y}$ and the landmark uncertainty covariance $\Sigma_{\mathbf{y}}$).

Let $\mathbf{z}_i = (x_i, y_i)$ be a candidate to the i^{th} landmark, being $\mathbf{y}^c_i$ the current landmark estimate, $\mathbf{\Omega}_{\mathbf{y}^c_i}$ a $L \times L$ patch centered at $\mathbf{y}^c_i$, a_i a binary variable that denotes correct landmark alignment, $\mathcal{D}_i$ the score of a generic local detector and $\mathbf{I}$ the target image up to a similarity transformation (typically the detector is designed to operate at a given scale). The probability of pixel $\mathbf{z}_i$ to be aligned is given by

$$p_i(\mathbf{z}_i) = p(a_i = 1|\mathbf{I}(\mathbf{z}_i), \mathcal{D}_i) = \frac{1}{1 + e^{-a_i \mathcal{D}_i(\mathbf{I}(\mathbf{z}_i))}} \tag{3.6}$$

where the detector score is converted to probability using the logistic function. The parameters $\mathbf{y}_i$ and $\Sigma_{\mathbf{y}_i}$ can be found by minimizing the expression [104]

$$\arg \min_{\mathbf{y}_i, \Sigma_{\mathbf{y}_i}} \sum_{\mathbf{z}_i \in \mathbf{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i) \mathcal{N}(\mathbf{z}_i | \mathbf{y}_i, \Sigma_{\mathbf{y}_i}) \tag{3.7}$$

where several strategies can be used to do this optimization.

Weighted Peak Response (WPR): The simplest solution is to take the spatial location where the response map has a higher score [98]. The new landmark position is then weighted by a factor that reflects the peak confidence. Formally, the WPR solution is given by

$$\mathbf{y}_i^{\text{WPR}} = \max_{\mathbf{z}_i \in \mathbf{\Omega}_{\mathbf{y}_i^c}} \left(p_i(\mathbf{z}_i)\right), \quad \Sigma_{\mathbf{y}_i}^{\text{WPR}} = diag(p_i(\mathbf{y}_i^{\text{WPR}})^{-1}) \tag{3.8}$$

that is equivalent to approximate each response map by an isotropic Gaussian $\mathcal{N}(\mathbf{z}_i | \mathbf{y}_i^{\text{WPR}}, \Sigma_{\mathbf{y}_i}^{\text{WPR}})$.

Gaussian Response (GR): The previous approach was extended in [115] to approximate the response maps by a full Gaussian distribution $\mathcal{N}(\mathbf{z}_i | \mathbf{y}_i^{\text{GR}}, \Sigma_{\mathbf{y}_i}^{\text{GR}})$. This is equivalent to fit a Gaussian density to weighted data. Let $d = \sum_{\mathbf{z}_i \in \mathbf{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i)$, the solution is given by

$$\mathbf{y}_i^{\text{GR}} = \frac{1}{d} \sum_{\mathbf{z}_i \in \mathbf{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i) \mathbf{z}_i, \quad \Sigma_{\mathbf{y}_i}^{\text{GR}} = \frac{1}{d-1} \sum_{\mathbf{z}_i \in \mathbf{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i)(\mathbf{z}_i - \mathbf{y}_i^{\text{GR}})(\mathbf{z}_i - \mathbf{y}_i^{\text{GR}})^T. \tag{3.9}$$

Kernel Density Estimator (KDE): The response maps can also be approximated by a nonparametric representation, namely using a Kernel Density Estimator (KDE) (isotropic Gaussian kernel with a bandwidth σ_h^2). Maximizing over the KDE is typically performed by using the well-known mean-shift algorithm [41]. The kernel bandwidth σ_h^2 is a free parameter that exhibits a strong influence on the resulting estimate. This problem can be addressed by an annealing bandwidth schedule. It can

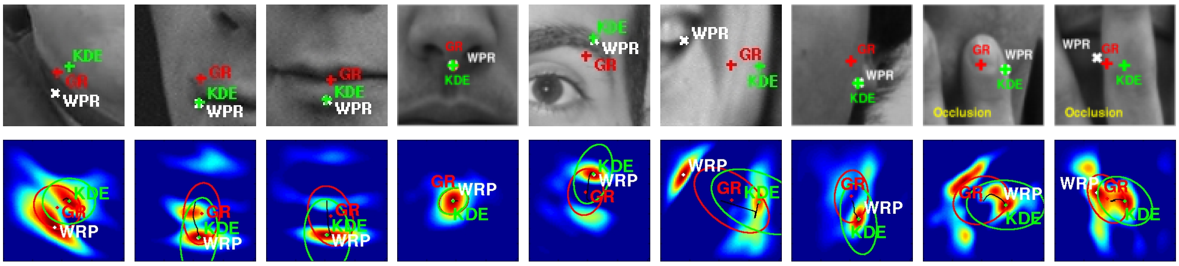


Figure 3.3: Qualitative comparison between the three local optimization strategies. The WPR simply chooses the maximum detector response. GR approximates the response map by a full Gaussian distribution. KDE uses the mean-shift algorithm to move to the nearest mode of the density. Its uncertainty covariance is found using the entire response map centered at the found mode. The two examples in the right show patches under occlusion (typically multimodal responses).

be shown [19] that there exists a σ_h^2 value such that the KDE is unimodal. As σ_h^2 is reduced, the modes divide and the smoothness of KDE decreases, guiding the optimization towards the true objective. Formally, the i^{th} annealed mean-shift landmark update is given by

$$\mathbf{y}_i^{\text{KDE}(\tau+1)} \leftarrow \frac{\sum_{\mathbf{z}_i \in \boldsymbol{\Omega}_{\mathbf{y}_i^c}} \mathbf{z}_i \; p_i(\mathbf{z}_i) \; \mathcal{N}(\mathbf{y}_i^{\text{KDE}(\tau)} | \mathbf{z}_i, \sigma_{h_j}^2 \mathbf{I}_2)}{\sum_{\mathbf{z}_i \in \boldsymbol{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i) \; \mathcal{N}(\mathbf{y}_i^{\text{KDE}(\tau)} | \mathbf{z}_i, \sigma_{h_j}^2 \mathbf{I}_2)} \tag{3.10}$$

where $\mathbf{I}_2$ is a two-dimensional identity matrix and $\sigma_{h_j}^2$ represents the decreasing annealed bandwidth. The KDE uncertainty error consists on computing the weighted covariance using the mean-shift results as mean

$$\Sigma_{\mathbf{y}_i}^{\text{KDE}} = \frac{1}{d-1} \sum_{\mathbf{z}_i \in \boldsymbol{\Omega}_{\mathbf{y}_i^c}} p_i(\mathbf{z}_i)(\mathbf{z}_i - \mathbf{y}_i^{\text{KDE}})(\mathbf{z}_i - \mathbf{y}_i^{\text{KDE}})^T. \tag{3.11}$$

Figure 3.3 highlights the differences between the three local optimization strategies (WPR, GR and KDE). Notice that DBASM deals with mild occlusions. When a landmark is under occlusion typically the response map is multi-modal. If a KDE local strategy is used (DBASM-KDE), the landmark update will select the nearest mode (eq.3.10) and the covariance of that landmark (eq.3.11) will be inherently large, modeling a high localization uncertainty. Then, the global optimization stage jointly combines all uncertainties (MAP sense), handling occlusions. Similarly, to deal with large occlusions, a minor tweak is required. One can simply set a large covariance for the occluded landmarks.

3.3.3 The Prior Term

The prior term, according to the approximations taken, can be written as

$$p(\mathbf{b}_k|\mathbf{b}_{k-1}) \propto \mathcal{N}(\mathbf{b}_k|\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) \tag{3.12}$$

where $\mu_{\mathbf{b}} = \mathbf{b}_{k-1}$ and $\Sigma_{\mathbf{b}} = \Lambda + \Xi$. The Λ is the shape parameters covariance (diagonal matrix with PCA eigenvalues) and Ξ is an additive dynamic noise covariance (that can be estimated offline).

3.3.4 The MAP Global Alignment

An important property of Bayesian inference is that, when the likelihood and the prior are Gaussian distributions the posterior is also Gaussian [17]. Following the Bayes' theorem for Gaussian variables, and considering $p(\mathbf{b}_k|\mathbf{b}_{k-1})$ a prior Gaussian distribution for $\mathbf{b}_k$ and $p(\mathbf{y}|\mathbf{b}_k)$ a likelihood Gaussian distribution, the posterior distribution takes the form ([17], pag 90).

$$p(\mathbf{b}_k|\mathbf{y}) \propto \mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{3.13}$$

$$\begin{aligned} \boldsymbol{\Sigma} &= (\Sigma_{\mathbf{b}}^{-1} + \Phi^T \Sigma_{\mathbf{y}}^{-1} \Phi)^{-1} && (3.14) \\ \boldsymbol{\mu} &= \boldsymbol{\Sigma}(\Phi^T \Sigma_{\mathbf{y}}^{-1} \mathbf{y} + \Sigma_{\mathbf{b}}^{-1} \mu_{\mathbf{b}}). && (3.15) \end{aligned}$$

Note that, the conditional distribution $p(\mathbf{y}|\mathbf{b}_k)$ has a mean that is a linear function of $\mathbf{b}_k$ and a covariance which is independent of $\mathbf{b}_k$. This could be a possible solution to the global alignment optimization [104]. However, in practice, this is a naive approach because it does not model the covariance of the latent variables, $\mathbf{b}_k$, which is crucial to account for the confidence in the current parameters estimate.

Second Order Global Alignment

The MAP global alignment solution can be inferred by a Linear Dynamical System (LDS). The LDS is the ideal technique to model the covariance of the latent variables and solve the naive approach limitations. The LDS is a simple approach that recursively computes the posterior probability using incoming Gaussian measurements and a linear model process, taking into account all the available measures (same requirements as our alignment problem). The state and measurement equations of the LDS, according to the PDM alignment problem, can be written as

$$\mathbf{b}_k = \mathbf{A}\mathbf{b}_{k-1} + q \tag{3.16}$$

$$\Delta\mathbf{y} = \Phi\mathbf{b}_k + r \tag{3.17}$$

where the current shape parameters $\mathbf{b}_k$ are the hidden state vector, $q \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{b}})$ is the additive dynamic noise, $\Delta\mathbf{y}$ is the observed shape deviation that are related to the shape parameters by the linear relation Φ (eq.3.1) and r is the additive measurement noise following $r \sim \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{y}})$. The previous shape estimated parameters $\mathbf{b}_{k-1}$ are connected to the current parameters $\mathbf{b}_k$ by an identity relation plus noise ($\mathbf{A} = \mathbf{I}_n$).

We highlight that the final step of the LDS derivation consists of a Bayesian inference step [17] (using the Bayes' theorem for Gaussian variables), where the likelihood term is given by eq.3.5 and the prior follows $\mathcal{N}(\mathbf{A}\mu^{\mathbf{F}}_{k-1}, \mathbf{P}_{k-1})$ where

$$\mathbf{P}_{k-1} = (\Lambda + \Xi) + \mathbf{A}\mathbf{\Sigma}^{\mathbf{F}}_{k-1}\mathbf{A}^T. \tag{3.18}$$

From these equations we can see that the LDS keep up to date the uncertainty on the current estimate of the shape parameters. The LDS recursively computes the mean and covariance of the posterior distributions of the form

$$p(\mathbf{b}_k|\mathbf{y}_k, \ldots, \mathbf{y}_0) \propto \mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}^{\mathbf{F}}_k, \mathbf{\Sigma}^{\mathbf{F}}_k) \tag{3.19}$$

with the posterior mean $\boldsymbol{\mu}^{\mathbf{F}}_k$ and covariance $\mathbf{\Sigma}^{\mathbf{F}}_k$ given by the LDS formulas:

$$\mathbf{K} = \mathbf{P}_{k-1}\Phi^T(\Phi\mathbf{P}_{k-1}\Phi^T + \Sigma_{\mathbf{y}})^{-1} \tag{3.20}$$

$$\boldsymbol{\mu}^{\mathbf{F}}_k = \mathbf{A}\boldsymbol{\mu}^{\mathbf{F}}_{k-1} + \mathbf{K}(\mathbf{y} - \Phi\mathbf{A}\boldsymbol{\mu}^{\mathbf{F}}_{k-1}) \tag{3.21}$$

$$\mathbf{\Sigma}^{\mathbf{F}}_k = (\mathbf{I}_n - \mathbf{K}\Phi)\mathbf{P}_{k-1}. \tag{3.22}$$

Finally, the optimal shape parameters that maximize eq.3.2 are given by $\boldsymbol{\mu}_k^{\mathbf{F}}$. In order to estimate the pose parameters, we also apply the LDS paradigm. The difference is that, in this case, the state vector is given by $\mathbf{q}$ and the observation matrix is Ψ. The algorithm 5 summarizes the proposed DBASM global optimization.

DBASM is a more powerful representation than the naive Bayesian approach (eqs. 3.14 and 3.15), propagating both state and uncertainty.

```
Precompute:
The parametric models ($\mathbf{s}_0$, $\Phi$, $\Psi$) and the MOSSE filters in the Fourier domain $\mathbf{H}_i^*$
Initial estimate of the shape/pose parameters and covariances $(\mathbf{b}_0, \mathbf{P}_0)$ / $(\mathbf{q}_0, \mathbf{Q}_0)$.
repeat
    Warp image $\mathbf{I}$ to the base mesh using the current pose parameters $\mathbf{q}_k$ [0.5ms]
    Generate current shape $\mathbf{s} = \mathcal{S}(\mathbf{s}_0 + \Phi\mathbf{b}_k, \mathbf{q}_k)$
    for Landmark $i = 1$ to $v$ do
        Evaluate the detectors response (MOSSE correlation $\mathcal{F}^{-1}\{\mathcal{F}\{(\mathbf{I})\} \odot \mathbf{H}_i^*\}$) [3ms]
        Find $\mathbf{y}_i$ and $\Sigma_{\mathbf{y}_i}$ using a local strategy (sec. 3.3.2),e.g. if using KDE, eqs.3.10 and 3.11, respectively.
    end
    Update the pose parameters and their covariance [0.1ms]:
        $\mathbf{Q}_{k-1} = (\Lambda_q + \Xi_q + \mathbf{Q}_{k-1})$,   $\mathbf{K}_q = \mathbf{Q}_{k-1}\Psi^T(\Psi\mathbf{Q}_{k-1}\Psi^T + \Sigma_{\mathbf{y}})^{-1}$
        $\mathbf{q}_k = \mathbf{q}_{k-1} + \mathbf{K}_q(\mathbf{y} - \Psi\mathbf{q}_{k-1})$,   $\mathbf{Q}_k = (\mathbf{I}_4 - \mathbf{K}_q\Psi)\mathbf{Q}_{k-1}$
    Update the shape parameters (with pose correction) and their covariance [0.2ms]:
        $\mathbf{P}_{k-1} = (\Lambda + \Xi + \mathbf{P}_{k-1})$,   $\mathbf{K}_b = \mathbf{P}_{k-1}\Phi^T(\Phi\mathbf{P}_{k-1}\Phi^T + \Sigma_{\mathbf{y}})^{-1}$
        $\mathbf{b}_k = \mathbf{b}_{k-1} + \mathbf{K}_b(\mathbf{y} - \Phi\mathbf{b}_{k-1} - \Psi\mathbf{q}_k)$,   $\mathbf{P}_k = (\mathbf{I}_n - \mathbf{K}_b\Phi)\mathbf{P}_{k-1}$
until $||\boldsymbol{b}_k - \boldsymbol{b}_{k-1}|| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 5: Overview of the DBASM method. The performance of DBASM is comparable to ASM [98], CQF [115] or SCMS [41] depending of the local strategy DBASM-WPR, DBASM-GR or DBASM-KDE, respectively. It achieves near real-time performance. The bottleneck is always obtaining the response maps (3ms x number landmarks), although it can be done in parallel.

3.3.5 Hierarchical Search (DBASM-KDE-H)

A slightly different annealing approach is proposed in this section. When the local response maps are approximated by KDE representations, the overall alignment can be done by a hierarchical search strategy. The standard search uses the mean-shift algorithm with an iterative kernel bandwidth relaxation, e.g. $\sigma^2_{h_j} = [15, 10, 5, 2]$, followed by a global optimization step (LDS MAP formulation). However, the mean-shift bandwidth annealing schedule can be combined with additional global optimization steps. Bottom levels use highest KDE bandwidth and perform global optimization steps. Then the next level shrinks the bandwidth and repeats the process. This solution is composed by multiple levels of fixed kernel bandwidth mean-shifts followed by global optimization steps (the annealing is performed between hierarchical levels).

Algorithms 6 and 7 highlight the differences between these two annealing schedules. The hierarchical version when compared to the standard search, forces more global optimization steps to take place, which in some cases produces better results (see later evaluation section 3.4.3).

```
repeat
    Warp image, generate shape.
    for Landmark $i = 1$ to $v$ do
        Detector response
        Find $\mathbf{y}_i$ and $\Sigma_{\mathbf{y}_i}$ using KDE
        for $\sigma^2_{h_j} = [15, 10, 5, 2]$ do
            Mean-Shift using $\sigma^2_{h_j}$
        end
        Global Optimization Step
    end
until $||update|| \leq \varepsilon$ / max iter ;
```

Algorithm 6: Standard KDE search.

```
for $\sigma^2_{h_j} = [15, 10, 5, 2]$ do
    repeat
        Warp image, generate shape.
        for Landmark $i = 1$ to $v$ do
            Detector response
            Find $\mathbf{y}_i$ and $\Sigma_{\mathbf{y}_i}$ using KDE
            Mean-Shift using $\sigma^2_{h_j}$
            Global Optimization Step
        end
    until $||update|| \leq \varepsilon$ / max iter ;
end
```

Algorithm 7: Hierarchical search.

3.4 Evaluation Results

The experiments were designed to evaluate the local detector (MOSSE [28]) and the new Bayesian global optimization (DBASM). All the experiments were conducted on several databases with publicly available ground truth. **(1)** The IMM [57] database that consists on 240 annotated images of 40 different human faces presenting different head pose, illumination, and facial expression (58 landmarks). **(2)** The BioID [61] dataset contains 1521 images, each showing a near frontal view of a face of one of 23 different subjects (20 landmarks). **(3)** The XM2VTS [45] database has 2360 images of frontal faces from 295 subjects (68 landmarks). **(4)** The tracking performance is evaluated on the FGNet Talking Face (TF) [32] video sequence that holds 5000 frames of video of an individual engaged in a conversation (68 landmarks). **(5)** Finally, a qualitative evaluation was also performed using the Labeled Faces in the Wild (LFW) [33] database that contains images taken under variability in pose, lighting, focus, facial expression, occlusions, different backgrounds, etc.

3.4.1 Local Detector - The MOSSE filter

The Minimum Output Sum of Squared Error (MOSSE) filter, recently proposed in [28], finds the optimal filter that minimizes the Sum of Squared Differences (SSD) to a desired correlation output. Briefly, correlation can be computed in the frequency domain as the element-wise multiplication of the 2D Fourier transform ($\mathcal{F}$) of an input image $\mathbf{I}$ with a filter $\mathbf{H}$, also defined in the Fourier domain as

$$\mathbf{G} = \mathcal{F}\{\mathbf{I}\} \odot \mathbf{H}^* \tag{3.23}$$

where the $\odot$ symbol represents the Hadamard product and $(*)$ is the complex conjugate. The correlation value is given by $\mathcal{F}^{-1}\{\mathbf{G}\}$, the inverse Fourier transform of $\mathbf{G}$.

MOSSE finds the filter $\mathbf{H}$, in the Fourier domain, that minimizes the SSD between the actual output of the correlation and the desired output of the correlation, across a set of N training images,

$$\min_{\mathbf{H}^*} \sum_{j=1}^{N} \left(\mathcal{F}\{\mathbf{I}_j\} \odot \mathbf{H}^* - \mathbf{G}_j\right)^2 \tag{3.24}$$

where $\mathbf{G}$ is obtained by sampling a 2D Gaussian uniformly. Solving for the filter $\mathbf{H}^*$ yields the closed form solution

$$\mathbf{H}^* = \frac{\sum_{j=1}^{N} \mathbf{G}_j \odot \mathcal{F}\{\mathbf{I}_j\}^*}{\sum_{j=1}^{N} \mathcal{F}\{\mathbf{I}_j\} \odot \mathcal{F}\{\mathbf{I}_j\}^*}. \tag{3.25}$$

The MOSSE filter maps all aligned training patch examples to an output, $\mathbf{G}$, centered at the feature location, producing notably stable correlation filters.

At the training stage, each patch example is normalized to have zero mean and a unitary norm, and is multiplied by a cosine window (required to solve the Fourier Transform periodicity problem). This also has the benefit of emphasizing the target center. These filters have a high invariance to illumination changes, due to their null DC component and revealed to be highly suitable to the task of generic face alignment (see figure 3.2).

3.4.2 Evaluating Local Detectors

Three landmark expert detectors were evaluated. The most used detector [115][41] is based on a linear classifier built from aligned (positive) and misaligned (negative) grey level patch examples (see image 3.4). The score of the i^{th} linear detector is given by

$$\mathcal{D}_i^{\text{linear}}(\mathbf{I}(\mathbf{y}_i)) = \mathbf{w}_i^T \mathbf{I}(\mathbf{y}_i) + b_i, \tag{3.26}$$

with $\mathbf{w}_i$ being the linear weight, b_i the bias constant and $\mathbf{I}(\mathbf{y}_i)$ a vectorized patch of pixel values sampled at $\mathbf{y}_i$. Similarly, a quadratic classifier can be used

$$\mathcal{D}_i^{\text{quadratic}}(\mathbf{I}(\mathbf{y}_i)) = \mathbf{I}(\mathbf{y}_i)^T \mathbf{Q}_i \mathbf{I}(\mathbf{y}_i) + \mathbf{L}_i^T \mathbf{I}(\mathbf{y}_i) + b_i \tag{3.27}$$

with $\mathbf{Q}_i$ and $\mathbf{L}_i$ being the quadratic and linear terms, respectively. Finally, the MOSSE filter correlation gives

$$\mathcal{D}_i^{\text{MOSSE}}(\mathbf{I}(\mathbf{y}_i)) = \mathcal{F}^{-1}\{\mathcal{F}\{\mathbf{I}(\mathbf{y}_i)\} \odot \mathbf{H}_i^*\} \tag{3.28}$$

where $\mathbf{H}_i^*$ is the MOSSE filter from eq.3.25. Both linear and quadratic classifiers (linear-SVM [77] and Quadratic Discriminant Analysis) were trained using images from the

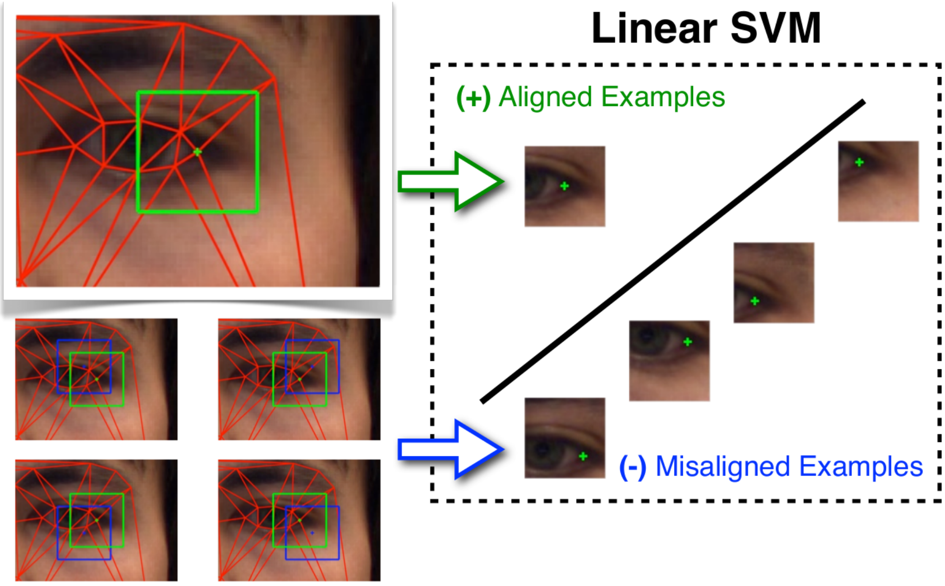


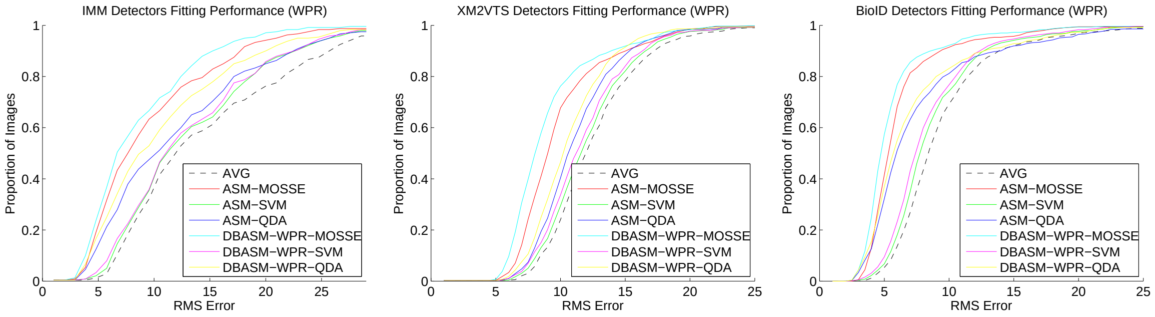
Figure 3.4: SVM based local landmark detectors. The expert local landmark detectors consists in train a linear SVM classifier with aligned (+ positive) versus misaligned (- negative) patch examples for each individual landmark.

IMM [57] dataset with 144 negative patch examples (for each landmark and each image) being misaligned up to 12 pixels in x and y translation.

The MOSSE filters were built using aligned patch samples with size 128 × 128. A power of two patch size is used to speed up the FFT computation, however only a 40 × 40 subwindow of the output is considered. During the MOSSE filter building, each training patch requires a normalization step. Each example is normalized to have a zero mean and a unitary norm and is multiplied by a cosine window. The desired output $\mathbf{G}$ (eq.3.25) is set to be a 2D Gaussian function centered at the landmark with 3 pixels of standard deviation.

The global optimization method that best evaluates the detectors performance is the approach that relies the most on the output of the detector, i.e., the Active Shape Models (ASM) [98]. The results are present in the form of fitting performance curves, which were also adopted by [22][24][23][115][41]. These curves show the percentage of faces that achieved convergence with a given Root Mean Square (RMS) error amount. The figure 3.5 shows fitting performance curves that compare the three kinds of detec-

WPR:



(a) IMM [57] database (b) XM2VTS [45] database (c) BioID [61] database

Figure 3.5: Fitting performance curves comparing different detectors (Linear, Quadratic and MOSSE) on the IMM, XM2VTS and BioID database, respectively. The AVG means the average location provided by the initial estimate (Adaboost [75] face detector).

tors using the ASM[98] optimization[1] and the proposed global DBASM technique using a Weighted Peak Response strategy (DBASM-WPR). From the results several conclusions can be highlighted: (1) the MOSSE filter always outperforms the others, specially when using simpler optimization methods; (2) the DBASM optimization improves the results even with simple detectors; (3) maximum performance can be achieved by using the MOSSE detector and the DBASM optimization.

The use of MOSSE filters is an interesting solution that works well in practice and is particularly suited to detection of facial parts. However it is important to stress that is not crucial for the performance of the Bayesian formulation. DBASM still improves performance when using standard detectors.

3.4.3 Evaluating Global Optimization Strategies

In this section the DBASM optimization strategy is evaluated w.r.t. state-of-the-art global alignment solutions. The proposed DBASM and DBASM-H methods are compared with (1) ASM [98], (2) CQF [115], (3) BCLM [104], (4) GMM [46] using 3

[1] The ASM [98], CQF [115] and SCMS [41] use as local optimizations the WPR, GR and KDE strategies, respectively.

Gaussians (GMM3) and (5) SCMS [41]. Note that the DBASM can be used with different local strategies to approximate the response maps (e.g. WPR, GR or KDE as described in section 3.3.2). In these experiments the KDE was fixed as local strategy (BCLM-KDE, SCMS-KDE, DBASM-KDE) in order to compare the global optimization approaches. The results from ASM, CQF and GMM3 are provided as a baseline. The same bandwidth schedule of $\sigma_h^2 = (15, 10, 5, 2)$ is always used for KDE. All the experiments, in this section, use MOSSE filters as local detectors (using the same settings as in section 3.4.2) built with only training images from the IMM [57] set and tested on the remaining datasets[2]. In all cases, the nonrigid parameters start from zero, the similarity parameters were initialized by a face detection [75] and the model was fitted until convergence (limited to a maximum of 20 iterations).

Figure 3.6 shows the fitting performance curves for the IMM, XM2VTS and BioID datasets, respectively. The CQF performs better than GMM3, mainly because GMM is very prone to local optimums due to its multimodal nature (it is worth mentioning that given a good initial estimate GMM offers a superior fitting quality). The main drawback of CQF is the limited accuracy due to the over-smoothness of the response map (see figure 3.3). The BCLM is slightly better than SCMS due to its improved parameter update (MAP update vs first order forwards additive). The SCMS improves the results when compared to CQF due to the high accuracy provided by the mean-shift. In some cases, the ASM achieves a comparable performance to the SCMS; the reason for this relies on the excellent performance of the MOSSE detector. The proposed Bayesian global optimization (DBASM) outperforms all previous methods, by modeling the covariance of the latent variables which represent the confidence in the current parameters estimate (see figure 3.6). The results show that the hierarchical annealing version of DBASM-KDE (DBASM-KDE-H) performs slightly better, but at the cost of more iterations.

Figure B.1, in appendix B, shows detailed fitting performance curves arranged by comparable local optimization strategies (WPR, GR and KDE). Additionally, the same

[2] The results presented on the IMM dataset use training images collected at our institution. This is done due to incompatibility of the annotation formats.

overall evaluation is repeated but now using the linear SVM as local feature detectors. The results are shown in figure B.2 with the corresponding table B.1 of quantitative results. The overall conclusions remain the same, DBASM still improves performance when using standard detectors.

Tracking Performance

Tracking performance is also tested on the FGNET Talking Face video sequence (figure 3.7). Each frame is fitted using as initial estimate the previously estimated shape and pose parameters. Figure B.3, in the appendix B, show the equivalent results when using the SVM linear detectors. The relative performance between the global optimization approaches is similar to the previous experiments, where the DBASM technique yields the best performance.

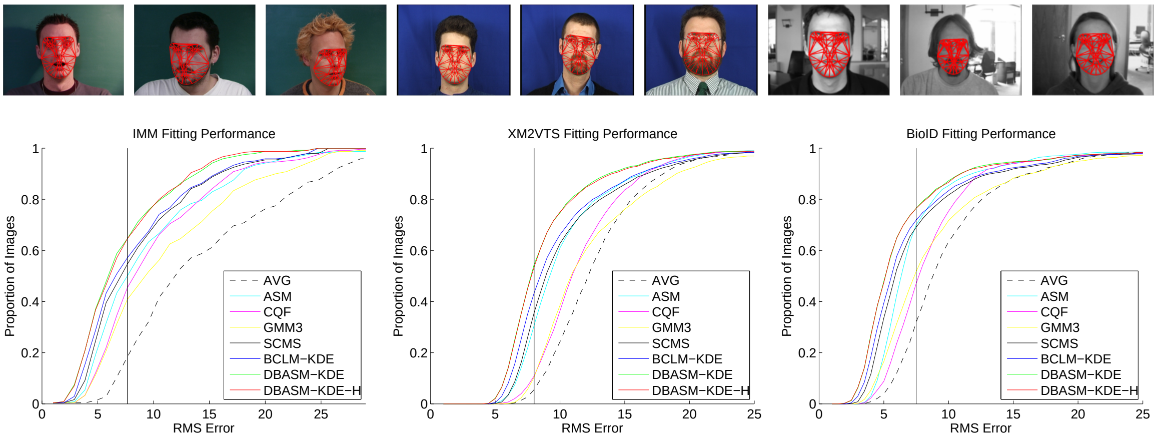
Qualitative evaluation is also performed using the challenging Labeled Faces in the Wild (LFW) database [33], where some results can be seen on figure 3.8.

3.4.4 Evaluating Multiple Feature Detectors

This formulation allows different patch alignment detectors to be seamlessly incorporated into the model. Multiple shape observations measurements can be considered by just updating the posterior distribution $\mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}_k^{\mathbf{F}}, \boldsymbol{\Sigma}_k^{\mathbf{F}})$ using multiple times the LDS correction steps.

In this section a global fitting strategy (DBASM-GR) is set and used to evaluate the fitting performance when using multiple landmark local detectors. To make a fair comparison the same kind of detector is used, in this case a linear SVM build from aligned (positive) and misaligned (negative) examples. Using the same settings, described in section 3.4.2, three independent linear SVM detectors were trained, one uses as input features the grey level patch values $\mathbf{I}(\mathbf{z})$, the second uses the magnitude of the gradients $\sqrt{\mathbf{I}_x(\mathbf{z})^2 + \mathbf{I}_y(\mathbf{z})^2}$ and the last one uses the orientation (phase) of the gradients $\arctan(\frac{\mathbf{I}_y(\mathbf{z})}{\mathbf{I}_x(\mathbf{z})})$.

Figure 3.9 shows fitting curves for the DBASM-GR fitting approach when using just the first detector, Bayesian fusion of the first and the second detector and fusion of all



(a) IMM [57] database

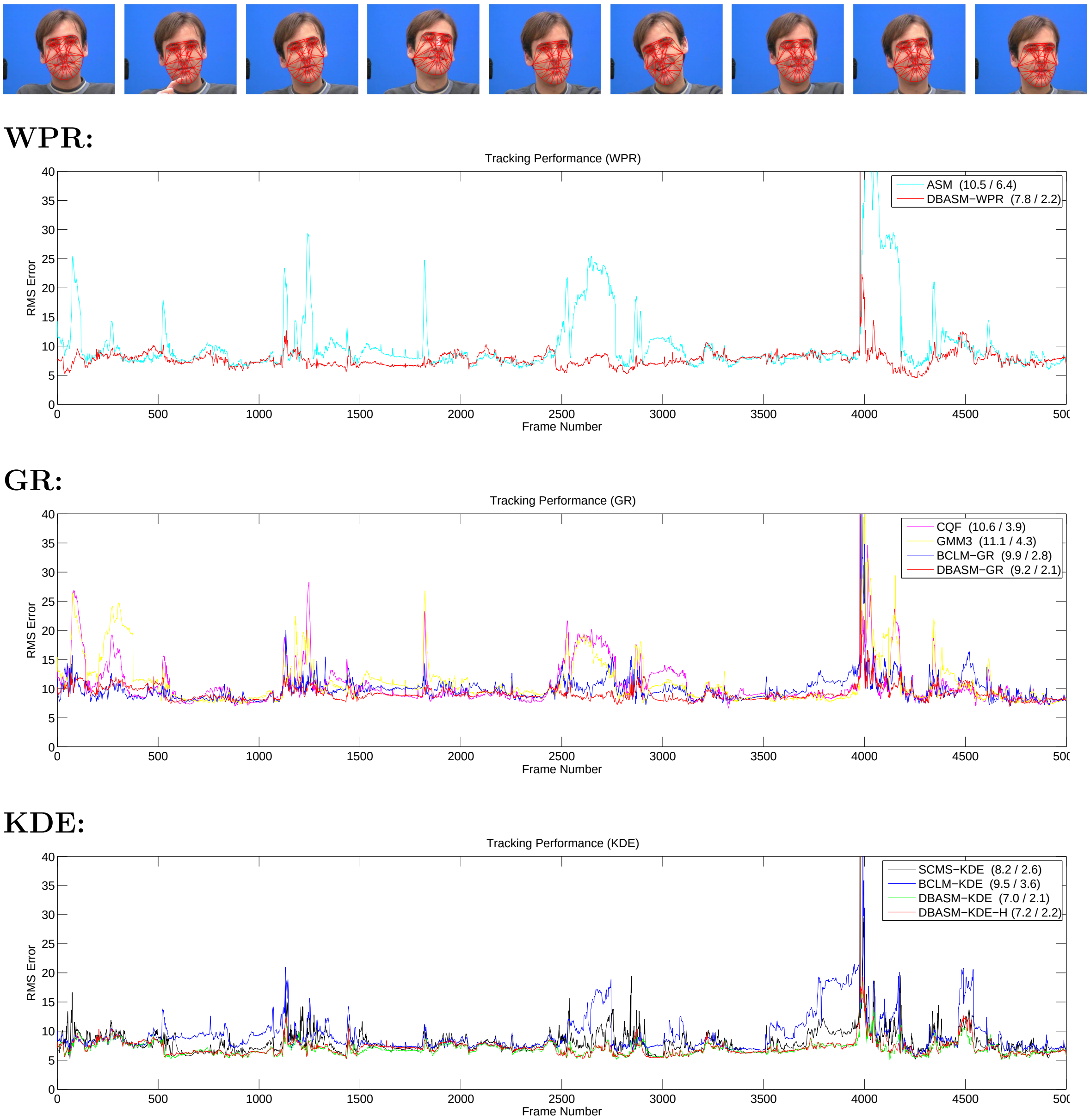
(b) XM2VTS [45] database

(c) BioID [61] database

Reference 7.5 RMS	IMM (240 images)		XM2VTS (2360 images)		BioID (1521 images)	
ASM	50.0		30.7		70.0	
DBASM-WPR* (our method)	**56.7**	(+6.7)	**45.1**	(+14.4)	**75.4**	(+5.4)
CQF	45.4		10.9		47.0	
GMM3	40.8	(-4.6)	10.4	(-0.5)	51.7	(+4.7)
BCLM-GR*	48.3	(+2.9)	15.9	(+5.0)	54.2	(+7.2)
DBASM-GR* (our method)	**50.4**	(+5.0)	**18.0**	(+7.1)	**62.2**	(+15.2)
SCMS-KDE	54.6		35.7		69.0	
BCLM-KDE	57.1	(+2.5)	43.4	(+7.7)	71.9	(+2.9)
DBASM-KDE (our method)	**64.6**	(+10.0)	**54.5**	(+18.8)	**76.5**	(+7.5)
DBASM-KDE-H (our method)	**64.6**	(+10.0)	53.5	(+17.8)	**76.5**	(+7.5)

The fitting curves for methods (*) are present in Appendix B.

Figure 3.6: Fitting performance curves. The table shows quantitative values taken by setting a fixed RMS error amount (7.5 pixels - vertical line in the graphics). Each table entry show how many percentage of images converge with less or equal RMS error than the reference. The results show that our proposed methods outperform all the other (using all the local strategies WPR, GR and KDE). Top images show DBASM-KDE fitting examples from each database.

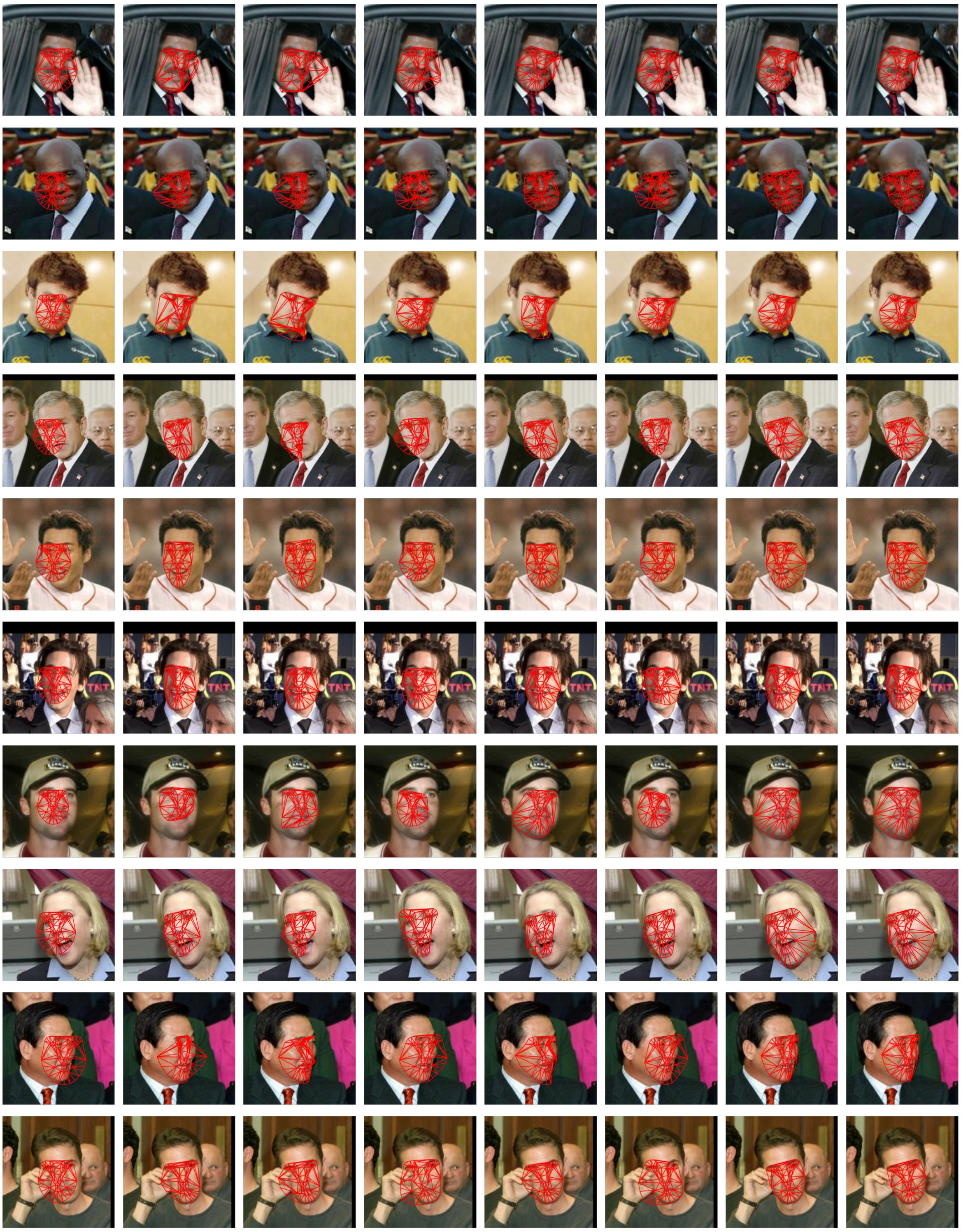


WPR:

GR:

KDE:

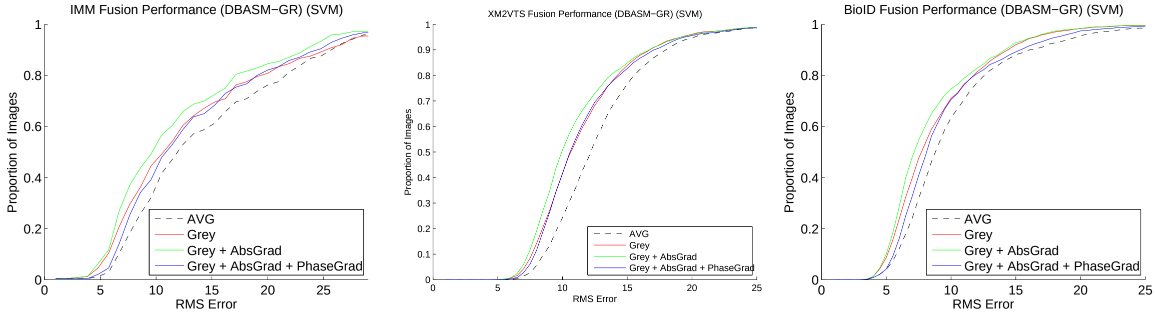
Figure 3.7: Evaluation of the tracking performance of several fitting algorithms on the FGNET Talking Face [32] sequence. The values on legend box are the mean and standard deviation RMS errors, respectively. Top images show DBASM-KDE fitting examples of the tested sequence. Best viewed in color. This evaluation can be seen at http://www.isr.uc.pt/~pedromartins/Videos/PhD.



(a) AVG (b) ASM (c) CQF (d) GMM3 (e) BCLM - KDE (f) SCMS - KDE (g) DBASM - KDE (h) DBASM - KDE-H

Figure 3.8: Qualitative fitting results on LFW [33] database. The AVG means the initial mesh estimate.

DBASM-GR:



(a) IMM [57] database (b) XM2VTS [45] database (c) BioID [61] database

Figure 3.9: Fitting performance curves for DBASM-GR evaluating Bayesian fusion of multiple local SVM detectors. Three linear SVM detectors were trained. One uses grey level values as input features, other uses magnitude of the gradients and the last uses the phase of the gradients. The *'Grey'* label means using DBASM-GR with a single detector, *'Grey + AbsGrad'* means the fusion of two detectors and *'Grey + AbsGrad + PhaseGrad'* fusion of all three detectors.

the detectors. The results show that the fitting performance can be increased by using Bayesian fusion of the first and second detectors. However, 'bad' (noisy) detectors, like the phase of the gradients, penalize performance.

3.5 Modeling the Prior Term

This section presents an extension to the previously DBASM [71] global alignment strategy in section 3.3. The remain of the chapter corresponds to the paper: 'Discriminative Face Alignment using Conjugate Priors' [72] presented at the BMVC 2012.

Faces are nonrigid structures described by continuous dynamic transitions. In the Bayesian paradigm the prior term can be used to encode the underlying dynamic of the shape. The prior term follows a Gaussian distribution with mean $\mu_{\mathbf{b}}$ and covariance $\Sigma_{\mathbf{b}}$

$$p(\mathbf{b}_k|\mathbf{b}_{k-1}) \propto \mathcal{N}(\mathbf{b}_k|\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}). \tag{3.29}$$

Mean $\mu_{\mathbf{b}}$ and covariance $\Sigma_{\mathbf{b}}$ of the data are assumed to be unknown and modeled as random variables ([1] pag.87-88). Recursive Bayesian estimation can be applied to infer the parameters of the prior distribution in eq.3.29. Defining $\mathbf{b}$ as an observable vector, the Bayes theorem tells us that the joint posterior density can be written as

$$p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}|\mathbf{b}) \propto p(\mathbf{b}|\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}})p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}). \tag{3.30}$$

Performing recursive Bayesian estimation with new observations requires that joint prior density $p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}})$ should have the same functional form than the joint posterior density $p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}|\mathbf{b})$. The joint prior density, conditioning on the covariance $\Sigma_{\mathbf{b}}$, can be written as

$$p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) = p(\mu_{\mathbf{b}}|\Sigma_{\mathbf{b}})p(\Sigma_{\mathbf{b}}). \tag{3.31}$$

The previous condition is true if we assume that the covariance follow an inverse-Wishart distribution and $\mu_{\mathbf{b}}|\Sigma_{\mathbf{b}}$ follow a normal distribution (the conjugate prior for a Gaussian with known mean is an inverse-Wishart distribution [1])

$$\Sigma_{\mathbf{b}} \sim \text{Inv-Wishart}_{\upsilon_{k-1}}(\Lambda^{-1}_{\upsilon_{k-1}}), \qquad \mu_{\mathbf{b}}|\Sigma_{\mathbf{b}} \sim \mathcal{N}(\theta_{k-1}, \frac{\Sigma_{\mathbf{b}}}{\kappa_{k-1}}) \tag{3.32}$$

where υ_{k-1} and Λ_{k-1} are the degrees of freedom and scale matrix for the inverse-Wishart distribution, respectively. θ_{k-1} is the prior mean and κ_{k-1} is the number of prior measurements. According with these assumptions, the joint prior density becomes

$$p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}) \propto |\Sigma_{\mathbf{b}}|^{-(\upsilon_{k-1}+n)/2+1} \exp\left(-\frac{1}{2}\mathrm{tr}(\Lambda_{k-1}\Sigma_{\mathbf{b}}^{-1}) - \frac{\kappa_{k-1}}{2}(\mu_{\mathbf{b}} - \theta_{k-1})^T\Sigma_{\mathbf{b}}^{-1}(\mu_{\mathbf{b}} - \theta_{k-1})\right), \tag{3.33}$$

a normal-inverse Wishart distribution (the product between a Gaussian and an inverse-Wishart). We recall that n is the number of shape parameters.

The inference step in eq.3.30 involves a Gaussian likelihood and the joint prior $p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}})$, resulting in a joint posterior density of the same family (conjugate prior for a Gaussian with unknown mean and covariance), i.e. following a normal inverse-Wishart$(\theta_k, \Lambda_k/\kappa_k; \upsilon_k, \Lambda_k)$ distribution with the hyperparameters [1]:

$$\upsilon_k = \upsilon_{k-1} + m, \quad \kappa_k = \kappa_{k-1} + m \tag{3.34}$$

$$\theta_k = \frac{\kappa_{k-1}}{\kappa_{k-1}+m}\theta_{k-1} + \frac{m}{\kappa_{k-1}+m}\overline{\mathbf{b}} \tag{3.35}$$

$$\Lambda_k = \Lambda_{k-1} + \sum_{i=1}^{m}(\mathbf{b}_i - \overline{\mathbf{b}})(\mathbf{b}_i - \overline{\mathbf{b}})^T + \frac{\kappa_{k-1}m}{\kappa_{k-1}+m}(\overline{\mathbf{b}} - \theta_{k-1})(\overline{\mathbf{b}} - \theta_{k-1})^T \tag{3.36}$$

where $\overline{\mathbf{b}}$ is the mean of the new samples, m the number of samples used to update the model. The posterior mean θ_k is a weighted average between the prior mean θ_{k-1} and the sample mean $\overline{\mathbf{b}}$. The posterior degrees of freedom are equal to prior degrees of freedom plus the sample size. In the present case, the second term in eq.3.36 ($\sum_{i=1}^{M}\cdots$) is null because the model is updated with one sample each time ($m = 1$).

Marginalizing over the joint posterior distribution $p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}|\mathbf{b})$ (eq.3.30) with respect to $\Sigma_{\mathbf{b}}$ gives the marginal posterior distribution for the mean of the form

$$p(\mu_{\mathbf{b}}|\mathbf{b}) \propto t_{\upsilon_k-n+1}(\mu_{\mathbf{b}}|\theta_k, \Lambda_k/(\kappa_k(\upsilon_k - n + 1))). \tag{3.37}$$

where t_{υ_k-n+1} is the multivariate Student-t distribution with $\upsilon_k - n + 1$ degrees of freedom.

Using the expectation of marginal posterior distribution $p(\mu_{\mathbf{b}}|\mathbf{b})$ as the model parameters at time k, we get (see table of expectation for multivariate t-distributions e.g.[1] pag.576).

$$\mu_{\mathbf{b}_k} = E(\mu_{\mathbf{b}}|\mathbf{b}) = \theta_k. \tag{3.38}$$

Similarly, marginalizing over the joint posterior distribution $p(\mu_{\mathbf{b}}, \Sigma_{\mathbf{b}}|\mathbf{b})$ with respect to $\mu_{\mathbf{b}}$ gives the marginal posterior distribution $p(\Sigma_{\mathbf{b}}|\mathbf{b})$ that follows an inverse

Wishart distribution. The expectation for marginal posterior covariance is (see table of expectation for inverse Wishart distributions e.g.[1] pag.575)

$$\Sigma_{\mathbf{b}_k} = E(\Sigma_{\mathbf{b}}|\mathbf{b}) = (\upsilon_k - n - 1)^{-1}\Lambda_k. \tag{3.39}$$

3.5.1 MAP Global Alignment

In Bayesian inference, when the likelihood and the prior are Gaussian distributions the posterior is also a Gaussian. Consequently, a possible solution to the global alignment, can be given by the Bayes' theorem for Gaussian variables ([17], pag.90), considering $p(\mathbf{b}_k|\mathbf{b}_{k-1})$ a prior Gaussian distribution for $\mathbf{b}_k$ and $p(\mathbf{y}|\mathbf{b}_k)$ a likelihood Gaussian distribution. Note that, the conditional distribution $p(\mathbf{y}|\mathbf{b}_k)$ has a mean that is a linear function of $\mathbf{b}_k$ and a covariance which is independent of $\mathbf{b}_k$ (eq.3.4). However, we further extend this result by adding two main components: **(1)** use a second order estimate of the latent variables [71] (the covariance Σ_{k-1}). Using the covariance of the latent variables is a crucial issue, as it allows to account for the confidence on the current estimate (i.e. the amount of uncertainty in $\mathbf{b}_{k-1}$ should be considered in the estimate of $\mathbf{b}_k$). **(2)** Bayesian fusion of detectors. Allow to multiple (M) local detectors ($\sum_{m=1}^{M}\cdots$) to be seamlessly incorporated into the model, usually increase the fitting accuracy. The recursive posterior distribution takes the form of

$$p(\mathbf{b}_k|\mathbf{y}_k,\ldots,\mathbf{y}_0) \propto \mathcal{N}(\mathbf{b}_k|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{3.40}$$

$$\boldsymbol{\Sigma}_k = \left((\Sigma_{\mathbf{b}_k} + \Sigma_{k-1})^{-1} + \Phi^T \sum_{m=1}^{M}\left(\Sigma_{\mathbf{y}_{(m)}}^{-1}\right)\Phi\right)^{-1} \tag{3.41}$$

$$\boldsymbol{\mu}_k = \boldsymbol{\Sigma}_k\left(\Phi^T \sum_{m=1}^{M}\left(\Sigma_{\mathbf{y}_{(m)}}^{-1}\Delta\mathbf{y}_{(m)}\right) + (\Sigma_{\mathbf{b}_k} + \Sigma_{k-1})^{-1}\mu_{\mathbf{b}_k}\right) \tag{3.42}$$

where $\Delta\mathbf{y}_{(m)}$, $\Sigma_{\mathbf{y}_{(m)}}$ are the multiple likelihood observations.

The pose parameters $\mathbf{q}$ are estimated in the same way. The parameters of the normal inverse-Wishart distribution (eqs.3.34, 3.35 and 3.36) are kept up date and the global optimization step is used. However, the term Φ must be changed by Ψ (section 3.2) in both eqs.3.41 and 3.42. See algorithm 8 where the overall global optimization is summarized.

```
Precompute: The PDM $\mathbf{s}_0$, $\Phi$, $\Psi, \Lambda_{PCA} = \text{diag}(\lambda_1, \ldots, \lambda_n)$, where $\lambda_i$ is the $i^{th}$ PCA
eigenvalue and the local detectors $\mathbf{H}_i^*$
Initial estimate of the shape/pose parameters and their covariances $(\mathbf{b}_0, \Sigma_0)$   $(\mathbf{q}_0, \Sigma_0^q)$
  (shape: $\upsilon_0 = 2n$, $\kappa_0 = 1$, $\theta_0 = \mathbf{b}_0$, $\mathit{\Lambda}_0 = n\Lambda_{PCA}$)
  (pose: $\upsilon_0^q = 8$, $\kappa_0^q = 1$, $\theta_0^q = \mathbf{q}_0$, $\mathit{\Lambda}_0^q = 4\times\text{diag}([0.05\ 0.005\ 5\ 5]^2)$)
repeat
    Warp image $\mathbf{I}$ to the base mesh using the current pose parameters $\mathbf{q}_k$ [0.5ms]
    Generate current shape $\mathbf{s} = \mathcal{S}(\mathbf{s}_0 + \Phi\mathbf{b}_k, \mathbf{q}_k)$
    for  $Landmark\ i = 1$ to $v$ do
        Evaluate the $M$ detector(s) response(s), eq.3.25 [$M$ x 3ms]
        Find the likelihood parameters $\mathbf{y}_i$ and $\Sigma_{\mathbf{y}_i}$ using a local strategy (section 3.3.2)
    end
    Estimate the pose parameters: (shape observation: $\Delta\mathbf{y} = \mathbf{y} - \mathbf{s}_0$) [0.15ms]
      - Update parameters of the normal inv-Wishart distrib. using eqs.3.34, 3.35 and 3.36
      - Expectation of the prior parameters $\mu_{\mathbf{q}_k} = \theta_k^q$ and $\Sigma_{\mathbf{q}_k} = (\upsilon_k^q - 4 - 1)^{-1}\mathit{\Lambda}_k^q$
      - Evaluate the pose parameters $\mathbf{q}_k$ and the covariance $\Sigma_{\mathbf{q}_k}$ by eqs.3.42 and 3.41,
    (changing $\Phi$ by $\Psi$)
    Estimate the shape parameters: (shape observation: $\Delta\mathbf{y} = \mathbf{y} - \mathbf{s}_0 - \Psi\mathbf{q}_k$) [0.25ms]
      - Update parameters of the normal inv-Wishart distrib. using eqs.3.34, 3.35 and 3.36
      - Expectation of the prior parameters $\mu_{\mathbf{b}_k} = \theta_k$ and $\Sigma_{\mathbf{b}_k} = (\upsilon_k - n - 1)^{-1}\mathit{\Lambda}_k$
      - Evaluate the shape parameters $\mathbf{b}_k$ and the covariance $\Sigma_{\mathbf{b}_k}$ by eqs.3.42 and 3.41
until $||\boldsymbol{b}_k - \boldsymbol{b}_{k-1}|| \leq \varepsilon$ or maximum number of iterations reached ;
```

Algorithm 8: Overview of the Bayesian Active Shape Models (BASM) method. The performance of BASM is comparable to ASM [98], CQF [115] or SCMS [41] depending of the local strategy BASM-WPR, BASM-GR or BASM-KDE, respectively. It achieves near real-time performance. The bottleneck is always obtaining the response maps (M x 3ms x number landmarks), although it can be done in parallel.

Hierarchical Search (BASM-KDE-H)

According to section 3.3.5 when the local response maps are approximated by KDE representations, the global alignment can also be done by a hierarchical search. The same can be said for the BASM-KDE approach. The following sections refer to this method as BASM-KDE-H.

3.6 Evaluation Results

According to section 3.4 a similar evaluation protocol was followed in this section. The main assessment experiments were performed in the IMM [57], the BioID [61] and the XM2VTS [45] databases. Tracking performance was evaluated/compared using the FGNet Talking Face (TF) [32] video sequence and finally, qualitative results taken from the Labeled Faces in the Wild (LFW) [33] dataset are also shown.

Local Detectors

As previously discussed, performing a fair comparison requires that all the evaluated global optimization strategies use the same local detector. Section 3.4.2 shows that the recently proposed MOSSE filter [28] perform better than the most used detector: the linear classifier build from aligned (positive) and misaligned (negative) examples [115][41]. As so, all the following experiments use the MOSSE filter as local landmark detector. The MOSSE filters use the same settings than previous section 3.4.2, i.e. $\mathbf{I}_j$ (the aligned patch examples) have size of 128×128 and $\mathbf{G}$ (the desired output) is set to be a 2D Gaussian function centered at the landmark with 3 pixels of standard deviation (recall eq.3.25). In the following section, the performance of a Bayesian fusion of detections is also evaluated. The additional detector used is still a MOSSE filter but built from magnitude of gradients $||\nabla \mathbf{I}_j||$.

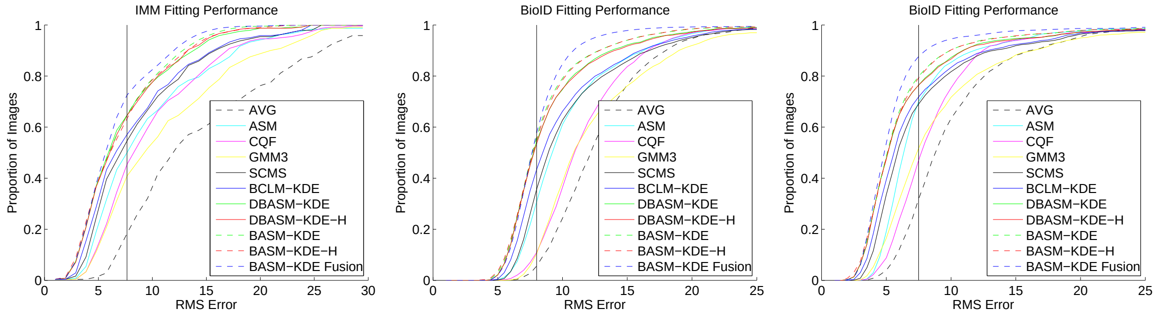
3.6.1 Evaluating Global Optimization Strategies

The BASM global optimization strategy (both BASM and BASM-H methods) are evaluated against (1) ASM [98], (2) CQF [115], (3) BCLM [104], (4) GMM [46] using three Gaussians (GMM3) (5) SCMS [41] and (6) DBASM [71] (described in previous section 3.3). Like previous section 3.4.3, the local search strategy KDE is fixed and comparable global optimization approaches are evaluated. The KDE kernel bandwidth schedule of $\sigma_h^2 = (15, 10, 5, 2)$ is used. Similarly, the results from ASM, CQF and GMM3 are provided as a baseline. In all cases, the nonrigid parameters start from zero, the similarity parameters were initialized by a face detection (Adaboost [75]) and the model was fitted until convergence (limited to a maximum of 20 iterations).

Figures 3.10 shows the fitting performance curves for the IMM, XM2VTS and BioID datasets, respectively. The table, in the same figure 3.10, shows quantitative values taken by sampling the curves using a fixed RMS error amount (7.5 pixels, shown as a vertical line in graphics). To avoid confusion, the remainder local strategies (WPR and GR) appear only in the table.

The results show that the proposed Bayesian global optimization (BASM) outperforms all previous methods. Explicitly modelling the prior distribution and using the covariance of the latent variables (inherited from DBASM) offers a significative increase in fitting performance. The Bayesian fusion of ($M = 2$) local detectors was evaluated using the method that previously achieved the best performance (BASM-KDE). The results (BASM-KDE Fusion) show that including multiple sets of patch alignment classifiers further improve ('a lot') the accuracy. In fact, this approach achieves the overall best results.

Tracking performance is evaluated in the FGNET Talking Face video sequence (figure 3.11). Each frame is fitted using as initial estimate the previously estimated shape and pose parameters. The relative performance between the global optimization approaches is similar to the previous experiments, where the BASM techniques yields the best performance. Here, the hierarchical annealing version of BASM-KDE (BASM-KDE-H) performs slightly better, but at the cost of more iterations. The fusion of local detectors (BASM-KDE Fusion), as expected, improves even further the performance.



(a) IMM [57] database (b) XM2VTS [45] database (c) BioID [61] database

Reference 7.5 RMS	IMM (240 images)		XM2VTS (2360 images)		BioID (1521 images)	
ASM	50.0		30.7		70.0	
DBASM-WPR (previous sec.3.3)	56.7	(+6.7)	45.1	(+14.4)	75.4	(+5.4)
BASM-WPR (this section)	**58.4**	(+8.4)	**47.4**	(+16.7)	**77.1**	(+7.1)
CQF	45.4		10.9		47.0	
GMM3	40.8	(-4.6)	10.4	(-0.5)	51.7	(+4.7)
BCLM-GR	48.3	(+2.9)	15.9	(+5.0)	54.2	(+7.2)
DBASM-GR (previous sec.3.3)	50.4	(+5.0)	18.0	(+7.1)	62.2	(+15.2)
BASM-GR (this section)	**51.8**	(+6.4)	**19.7**	(+8.8)	**63.5**	(+16.5)
SCMS-KDE	54.6		35.7		69.0	
BCLM-KDE	57.1	(+2.5)	43.4	(+7.7)	71.9	(+2.9)
DBASM-KDE (previous sec.3.3)	64.6	(+10.0)	54.5	(+18.8)	76.5	(+7.5)
DBASM-KDE-H (previous sec.3.3)	64.6	(+10.0)	53.5	(+17.8)	76.5	(+7.5)
BASM-KDE (this section)	**65.4**	(+10.8)	**57.0**	(+21.3)	**80.3**	(+11.3)
BASM-KDE-H (this section)	64.0	(+9.4)	56.6	(+20.9)	79.9	(+10.9)
BASM-KDE Fusion of 2 Detectors	**72.5**	(+17.9)	**58.7**	(+23.0)	**88.2**	(+19.2)

Figure 3.10: Fitting performance curves. The table shows quantitative values taken by setting a fixed RMS error amount (7.5 pixels - vertical line in the graphics). Each table entry show how many percentage of images converge with less (or equal) RMS error than the reference. The results show that our proposed methods outperform all the other (using all the local strategies WPR, GR and KDE). AVG is the location provided by the initial estimate [75].

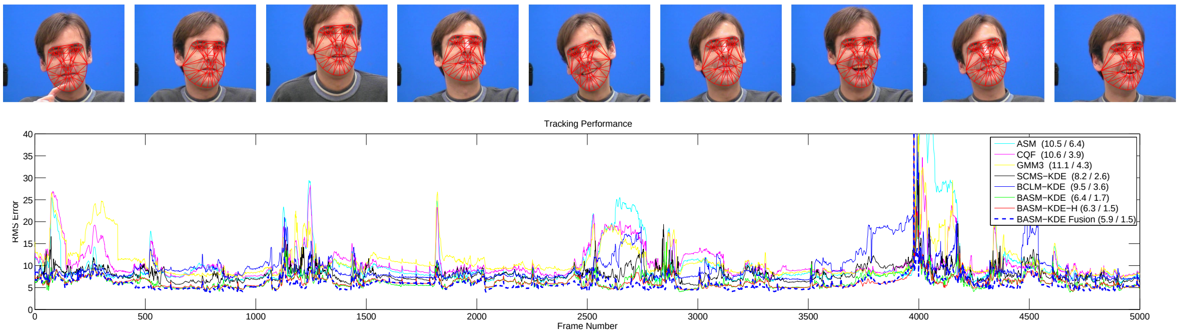


Figure 3.11: Evaluation of the tracking performance of several fitting algorithms on the FGNET Talking Face [32] sequence. The values on legend box are the mean and standard deviation RMS errors, respectively. Top images show BASM-KDE fitting examples. Best viewed in color. This evaluation can be seen at `http://www.isr.uc.pt/~pedromartins/Videos/PhD`.

Qualitative evaluation is also performed using the Labeled Faces in the Wild (LFW) database [33], where some results can be seen on figure 3.12.

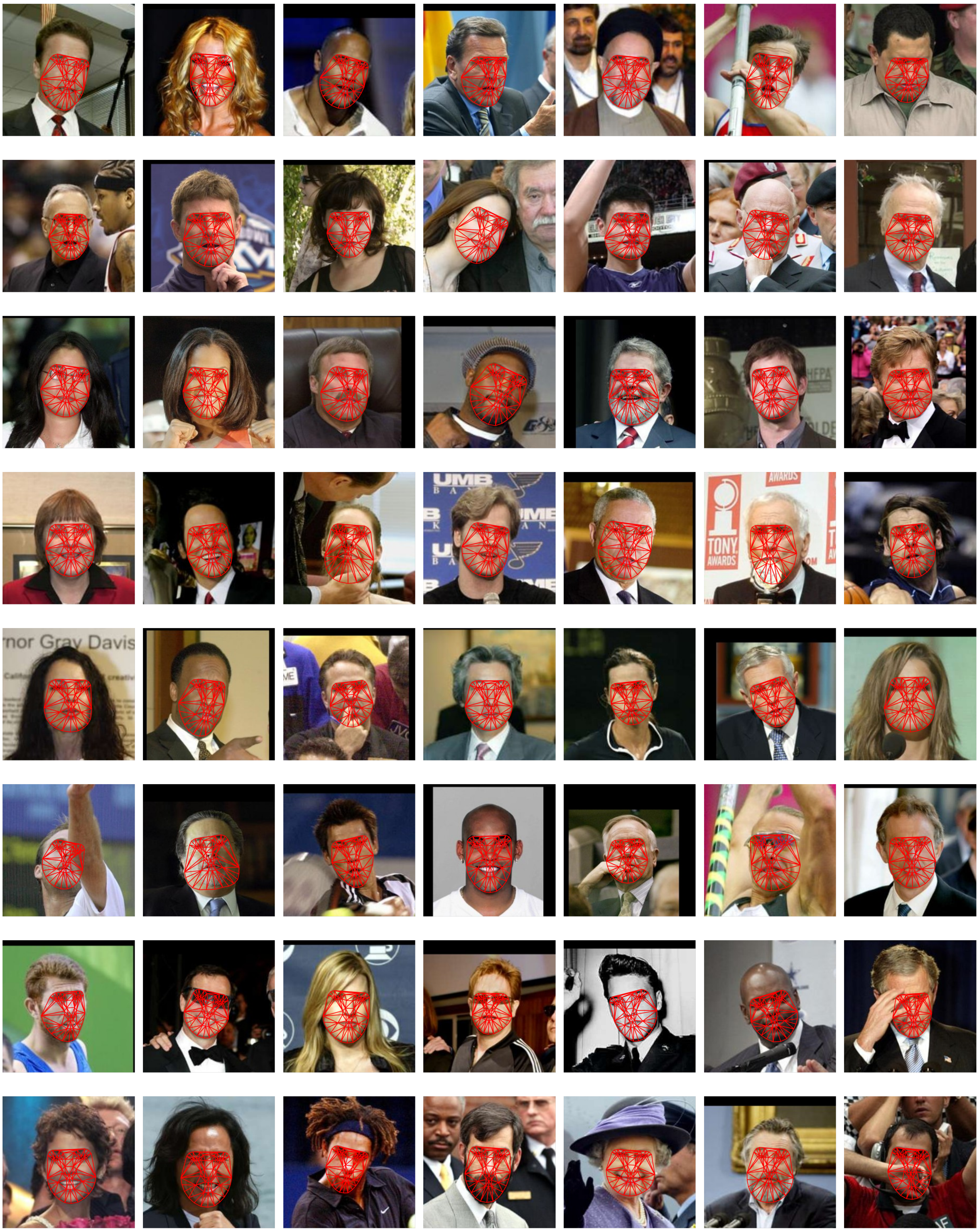


Figure 3.12: Qualitative BASM image alignment examples in the challenging Labeled Faces the Wild dataset [33]. See BASM qualitative video results at `http://www.isr.uc.pt/~pedromartins/Videos/PhD`.

3.7 Conclusions

An efficient solution to align facial parts in unseen images is described in this chapter. Fitting a Point Distribution Model (PDM) to an image involves a global optimization step where the responses of an ensemble of local feature detectors are jointly maximized. A novel Bayesian paradigm (DBASM) to solve the global alignment problem in a *maximum a posteriori* (MAP) sense is presented, being shown that the posterior distribution of the global warp can be efficiently inferred using a Linear Dynamical System (LDS). The main advantage w.r.t. previous formulations is that DBASM model the covariance of the latent variables which represent the confidence in the current parameters estimate.

The DBASM technique was extended to explicitly model the prior distribution. In this new strategy (BASM) the dynamic transitions of the PDM parameters, encoded by the prior distribution, were being continuously kept up to date. The extended global optimization strategy makes use of recursive Bayesian estimation to model a Gaussian prior, treating the mean and covariance as random variables. This means that not only the mean and the covariance are estimated, but also the probability distribution of the mean and the covariance.

Several performance evaluation results are presented, comparing both local detectors and global optimization strategies. Evaluating the local detectors show that the MOSSE correlation filters offer a superior performance in landmark local detection. Global optimizations evaluation were performed in several image publicly available datasets, namely on, the IMM, the XM2VTS, the BioID, and the Labeled Faces on the Wild. Tracking performance is also evaluated on a video sequence using the FGNET Talking Face dataset. The new Bayesian paradigms are shown to significantly outperform other state-of-the-art fitting solutions.

Chapter 4

Identity and Facial Expression Recognition

In this chapter, a solution for identity and facial expression recognition is proposed using a two stage classifier approach with a low dimensional representation of the geometry of the face. Face geometry is extracted from input images using the Active Appearance Models (AAM) and low dimensional manifolds were then derived using Laplacian EigenMaps (LE) resulting in two types of manifolds, one representing identity and the other person-specific facial expressions. The first stage uses a multiclass Support Vector Machines (SVM) to establish identity across expression changes. The second stage deals with person-specific expression recognition and is composed by a network of several Hidden Markov Models (HMM), each one specialized to a given facial emotion. The decision was made by the sequence that yielded the highest probability. For evaluation proposes a database was build consisting on 6770 images captured from four people exhibiting seven different emotions. The identity overall recognition rate was 96.8%. Facial expression results are identity dependent and the most expressive individual achieves 81.2% of overall recognition rate.

Publications

The contents of this chapter resulted in two main publications:

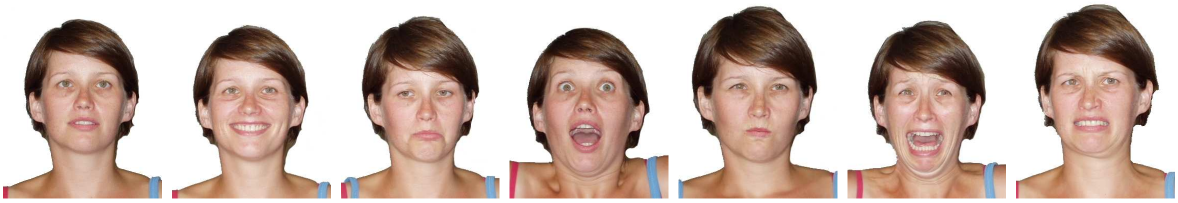
- Identity and Expression Recognition on Low Dimensional Manifolds [66]
  Pedro Martins, Jorge Batista
  ICIP 2009 - IEEE International Conference on Image Processing

- Simultaneous Identity and Expression Recognition Using Face Geometry on Low Dimensional Manifolds [67]
  Pedro Martins, Jorge Batista
  IbPria 2009 - Iberian Conference on Pattern Recognition and Image Analysis

4.1 Introduction

Facial expression is one of the most powerful, natural and immediate means for humans to share their emotions and intentions. However, automatic facial expression recognition is a difficult task because faces vary from one individual to another quite considerably due to differences in age, ethnicity, gender, occluding objects such as glasses and hair, pose and lighting changes. Psychological studies focus on the interpretation on this mean to interact and describe that there are six basic emotions universally recognized [21][62], namely: joy, sadness, surprise, fear, anger and disgust (see image 4.1). An automatic, efficient and accurate facial expression extraction system would thus be a powerful tool assisting on these studies, allowing also other kinds of applications such as Human Computer Interfaces (HCI), smart interactive systems, video compression, etc.

In the past, a lot of effort was dedicated to facial expression recognition in still images (static recognition). Many techniques have been applied such as Neural Networks [114], Gabor Wavelets [58] or the Active Appearance Models (AAM) [10][35][113][29]. More recently, attention has been shifted particularly towards modeling dynamical facial expressions. Facial dynamics is very important to efficiently classify emotions, since dynamic methods can deal with most ambiguities in static recognition by simply



(a) Neutral (b) Happy (c) Sad (d) Surprise (e) Anger (f) Fear (g) Disgust

Figure 4.1: The six basic human emotions universally recognized plus the neutral expression.

using previous time instants to estimate the most probable facial expression. Dynamical approaches can use shape deformations [29], texture dynamics [76] or a combination of them [112]. A dynamic classifier in [103] is based on building spatio-temporal model for each universal expression. The recognition of unseen expression uses the Hausdorff distance to compute dissimilarity values for classification. The authors in [112] propose a dynamic recognition based on the differential Active Appearance Model parameters. A sequence of input frames is fitted using the classical AAM then a specific frame is selected as reference frame. Then the corresponding sequence of differential AAM parameters is recognized by computing the directed Hausdorff distance and the K-Nearest Neighbor classifier. In [18], a Bayesian approach is used to modeling temporal transitions of facial expressions represented in a manifold using Local Binary Pattern (LBP) [102] features as facial appearance representation.

The identity and facial expression recognition approach presented in this chapter [66][67], is based on the idea that it is straightforward for a human to capture the emotion and consequently the identity of a mimic, or someone known using makeup. Humans can understand both the identity/expression based only on facial motion. This general idea suggests that face geometry could be used to recognize both the identity and facial expression (focusing on the six basic emotions plus the neutral one).

In this work, face images were represented by a set of 2D sparse feature points extracted using Active Appearance Models (AAM) [99][38]. AAMs are an effective way to locate facial features, modeling both shape and texture from an observed training set, being able to extract relevant face information without background interference. Both the identity and person-specific expression manifolds were learnt in a facial geometric feature space using Laplacian EigenMaps (LE) [51]. LE are nonlinear dimension reduction techniques that derive a low dimensional manifold lying in a higher dimensional more complex manifold. Such manifold is derived by embedding image data into a low dimensional space, where a image sequence is then represented as a trajectory in that feature space.

The recognition is based on a two stage cascade of classifiers. The first stage uses a multiclass Support Vector Machines (SVM) [106] that determines the identity. The second stage deals with the facial expression, being composed by a network of Hidden Markov Models (HMM) [48]. For an input image, the AAM fitting framework extracts facial geometric related features and projects them into the identity manifold. The first SVM stage predicts the identity and the respective person-specific model is loaded to stage two. Here the extracted features are projected into the expression manifold and the HMM based network decides the most likely facial expression.

4.1.1 Outline

This chapter is organized as follows: **Section** 4.2 gives an brief introduction to the 2D Active Appearance Models (AAM), the building and fitting of such model. **Section** 4.3 briefly describes how to derive low dimensional manifolds using the Laplacian Eigen-Maps (LE). Similarly, a briefly introduction of the Hidden Markov Models (HMM) is presented in **Section** 4.4. The **Section** 4.5 addresses to the proposed approach of recognizing the identity and the facial expression. Experimental results are presented in **Section** 4.6 and, finally, **Section** 4.7 summarizes the chapter.

4.2 Active Appearance Models (AAM)

Active Appearance Models (AAM) [99][38] are generative linear parametric models of shape and texture, commonly used to model faces. These adaptive template matching methods, learn offline the variability of shape and texture that is captured from a representative training set, being able to fully describe with photorealistic quality the trained faces as well as unseen. The following sections describe how to build such models and how to fit them into an image.

4.2.1 Parametric Models of Shape and Appearance

The shape of an AAM is defined by the vertex locations of a 2D triangulated mesh. Mathematically, the representation used for a single v-point shape is a $2v$ vector given by $\mathbf{s} = (x_1, \dots, x_v, y_1, \dots, y_v)^T$. The AAM training data consists of a set of annotated images with the shape mesh marked (usually by hand). The shapes are aligned to a common mean shape using a Generalized Procrustes Analysis (GPA) [15], removing location, scale and rotation effects. Principal Components Analysis (PCA) [56] are then applied to the aligned shapes, resulting on the linear parametric model

$$\mathbf{s} = \mathcal{S}\left(\mathbf{s}_0 + \sum_{i=1}^{n} p_i \mathbf{s}_i, \mathbf{q}\right) \tag{4.1}$$

where the new shapes $\mathbf{s}$ are synthesized by deforming the mean shape $\mathbf{s}_0 = (x_1^0, \dots, x_v^0, y_1^0, \dots, y_v^0)^T$ using a weighted linear combination of eigenvectors $\mathbf{s}_i$. The shape generation is controlled by two sets of parameters: the shape parameters $\mathbf{p}$ and the 2D pose parameters $\mathbf{q}$ ($\mathcal{S}(., \mathbf{q})$ represents a similarity transformation). The shape parameters $\mathbf{p}$ are a n dimensional vector which represents the eigen-shapes mixing weights with n being the number of eigenvectors that hold a user defined variance, typically 95%.

The 2D pose is represented by the similarity parameters vector as $\mathbf{q} = (s\cos(\theta) - 1, s\sin(\theta), t_x, t_y)^T$ where s, θ, t_x,t_y are the scale, rotation and translations w.r.t. the base mesh $\mathbf{s}_0$, respectively. As described in section 3.2, an additional matrix Ψ is defined to hold four special eigenvectors that linearly model the 2D pose [38].

Building a texture model, requires warping each training image so that the control points match those of the mean shape, $\mathbf{s}_0$. In order to prevent holes, the texture mapping is performed using the reverse map with bilinear interpolation correction. This texture mapping procedure is performed, using a piecewise affine warp, i.e. partitioning the convex hull of the mean shape by a set of triangles using the Delaunay triangulation. Each pixel inside a triangle is mapped into the correspondent triangle in the mean shape

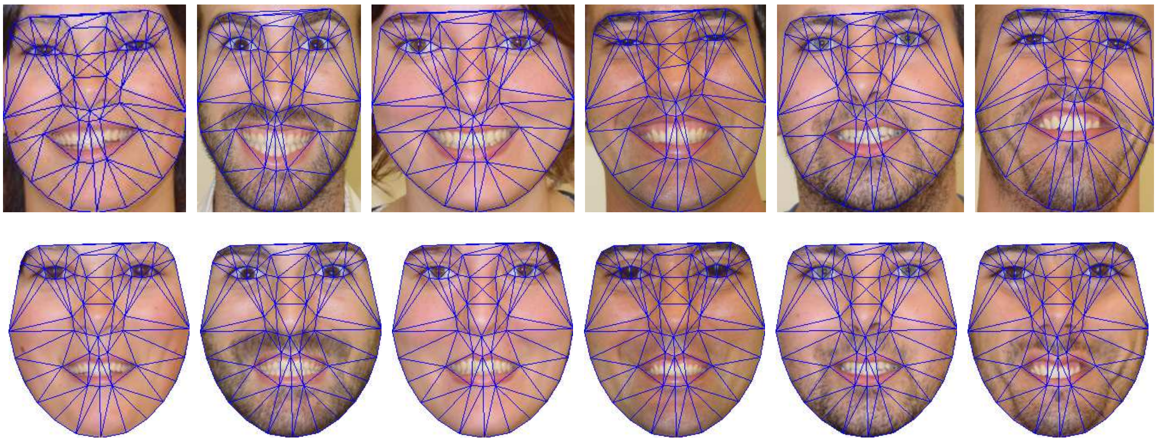


Figure 4.2: Texture warping examples. Top images $\mathbf{I}(\mathbf{x})$ are warped using $\mathbf{W}(\mathbf{x}, \mathbf{p}, \mathbf{q})$ into the images $\mathbf{I}(\mathbf{W}(\mathbf{x}, \mathbf{p}, \mathbf{q}))$ shown at the bottom.

using barycentric coordinates (see figure 4.2). This procedure removes differences in texture due shape changes, establishing a common texture reference frame. A texture model is obtained by applying a low-memory PCA on the normalized textures. Defining pixel coordinates as $\mathbf{x} = (x, y)^T$, the appearance of the AAM is an image, $\mathbf{A}(\mathbf{x})$, defined over the pixels $\mathbf{x} \in \mathbf{s}_0$ such as

$$\mathbf{A}(\mathbf{x}) = \mathbf{A}_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x}), \quad \mathbf{x} \in \mathbf{s}_0. \tag{4.2}$$

The appearance $\mathbf{A}(\mathbf{x})$ can be expressed as a base appearance $\mathbf{A}_0(\mathbf{x})$ plus a linear combination of m appearance images $\mathbf{A}_i(\mathbf{x})$ (EigenFaces). The coefficients λ_i are the appearance parameters.

4.2.2 Fitting an AAM into an Image

Fitting an AAM is usually formulated [38] as minimizing the texture error, in a least square sense, between the current model instance $\mathbf{A}(\mathbf{x})$ and the input backwarped image onto the base mesh $\mathbf{I}(\mathbf{W}(\mathbf{x}, \mathbf{p}, \mathbf{q}))$,

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x}\in\mathbf{s}_0} \left[\mathbf{A}_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \mathbf{A}_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}, \mathbf{p}, \mathbf{q})) \right]^2 . \tag{4.3}$$

In eq. 4.3 the warp $\mathbf{W}$ is the piecewise affine warp from the base mesh $\mathbf{s}_0$ to the current AAM shape $\mathbf{s}$. Hence, $\mathbf{W}$ is a function of the shape and pose parameters $\mathbf{p}$ and $\mathbf{q}$, respectively.

The Simultaneous Inverse Compositional (SIC) [84][85] minimize eq.4.3 by performing a Gauss-Newtow gradient descent optimization simultaneously on the warp parameters $\mathbf{p}$, the pose parameters $\mathbf{q}$ and the appearance parameters $\boldsymbol{\lambda}$.

Concatenating all the $n+4+m$ parameters in a single vector, $\mathbf{r} = (\mathbf{p}^T|\mathbf{q}^T|\boldsymbol{\lambda}^T)^T$ and denoting the Steepest Descent images [85] as

$$\mathbf{SD}_{\mathrm{SIC}}(\mathbf{x}) = \left(\nabla\mathbf{A}\frac{\partial\mathbf{W}}{\partial p_1}\cdots\nabla\mathbf{A}\frac{\partial\mathbf{W}}{\partial p_n}\nabla\mathbf{A}\frac{\partial\mathbf{W}}{\partial q_1}\cdots\nabla\mathbf{A}\frac{\partial\mathbf{W}}{\partial q_4}\mathbf{A}_1(\mathbf{x})\cdots\mathbf{A}_m(\mathbf{x})\right) \tag{4.4}$$

where $\nabla\mathbf{A}$ is defined as $\nabla\mathbf{A} = \nabla\mathbf{A}_0 + \sum_{i=1}^{m}\lambda_i\nabla\mathbf{A}_i$, the parameters update are computed as

$$\Delta\mathbf{r} = \mathbf{H}_{\mathrm{SIC}}^{-1}\sum_{\mathbf{x}\in\mathbf{s}_0}\mathbf{SD}_{\mathrm{SIC}}^T(\mathbf{x})\mathbf{E}(\mathbf{x}) \tag{4.5}$$

where $\mathbf{H}_{\mathrm{SIC}}$ is the Gauss-Newtow approximation of the Hessian given by

$$\mathbf{H}_{\mathrm{SIC}} = \sum_{\mathbf{x}\in\mathbf{s}_0}\mathbf{SD}_{\mathrm{SIC}}^T(\mathbf{x})\mathbf{SD}_{\mathrm{SIC}}(\mathbf{x}), \tag{4.6}$$

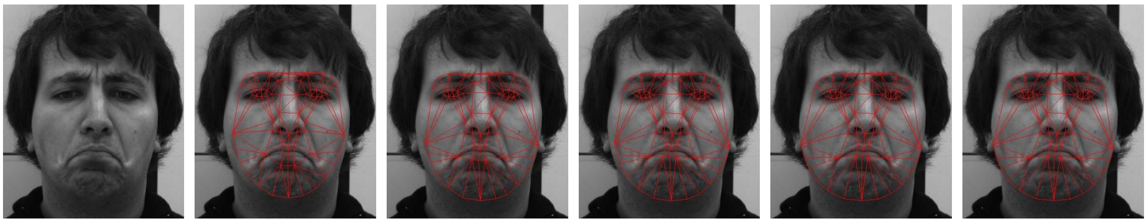
and the error image, $\mathbf{E}(\mathbf{x})$, is defined as the difference between the current model appearance and the most recent warped image

$$\mathbf{E}(\mathbf{x}) = \mathbf{A}_0(\mathbf{x}) + \sum_{i=1}^{m}\lambda_i\mathbf{A}_i(\mathbf{x}) - \mathbf{I}(\mathbf{W}(\mathbf{x},\mathbf{p},\mathbf{q})). \tag{4.7}$$

The Simultaneous Inverse Compositional, when compared with other fitting approaches, such as the Project-Out [38] or the precomputed numerical estimate [99], work rather slow, since the Steepest Descent images depend on the appearance parameters and they have to re-computed in every iteration. On the other hand, SIC achieves the better fitting accuracy which is desirable for our proposes.

4.3 Laplacian EigenMaps (LE)

Laplacian EigenMaps (LE) [51] is a nonlinear dimension reduction technique that derive a low dimensional manifold lying in a higher dimensional more complex manifold. The LE builds a graph that incorporates neighborhood information of the dataset and



(a) Input (b) 1^{st} (c) 2^{nd} (d) 5^{th} (e) 10^{th} (f) Final

Figure 4.3: AAM fitting example using SIC. The images show the evolution of the geometric model during several iterations until convergence.

using the notion of the Laplacian of the graph, computes a low dimensional representation that optimally preserves local neighborhood information. Given k feature points $\mathbf{x}_1, \cdots, \mathbf{x}_k \in \Re^n$, a weighted graph with k nodes is build, one for each point, with a set of edges connecting neighboring points. The embedding map is found by computing the eigenvectors of the graph Laplacian [51]. See algorithm 9 where this method is described. Finding such embedding map, Φ, requires tuning l nearest neighbors for graph building and select the number of dimensions, d, where the input features are projected into.

```
Build the Adjacency Graph:
Nodes $i$ and $j$ are connected by an edge to the $l$ nearest neighbors.
Choosing the weights $W_{ij}$: (if $i$ and $j$ are connected by an edge) then $W_{ij} = 1$
Build EigenMaps:
    $L\mathbf{f} = \lambda D\mathbf{f}$    (4.8)
    where $D_{ii} = \sum_j W_{ji}$ is a diagonal weight matrix and $L = D - W$ is the Laplacian
    matrix. Let $\mathbf{f}_0, \cdots, \mathbf{f}_{k-1}$ be the solutions of eq.4.8 order by eigenvalues
    $\lambda_0 = 0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$). Leaving out the eigenvector $\mathbf{f}_0$ corresponding to
    eigenvalue 0, the embedding $d$-dimensional Euclidean space is given by
    $\Phi = [\mathbf{f}_1 | \mathbf{f}_2 | \cdots | \mathbf{f}_d]$.
```

Algorithm 9: Laplacian EigenMaps (LE).

4.4 Hidden Markov Models (HMM)

Hidden Markov Models (HMM) [48] have been widely used for many classification and modeling problems. HMM is a finite set of *states*, each of which is associated with a multidimensional probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution. Only the observation is visible to an external observer, not the state, therefore states are hidden to the outside. An HMM is given by the following set of parameters:

1. The transition probabilities matrix **A** given by

$$\mathbf{A}_{i,j} = p(q_{t+1} = j | q_t = i), \qquad i, j = 1, \dots, h \tag{4.9}$$

   where q_t denotes the current state, and h the number of hidden states.

2. The probability distribution of each state

$$\mathbf{B}_j = p(O_t | q_t = j), \qquad j = 1, \dots, h \tag{4.10}$$

   with O_t being a observation at time t. The observations, in this case, use a continuous probability density function, usually approximated by a weighted mixture of M Gaussians as

$$\mathbf{B}_j(O_t) = \sum_{m=1}^{M} \omega_{jm} \mathcal{N}(O_t | \mu_{jm}, \Sigma_{jm}), \qquad j = 1, \dots, h \tag{4.11}$$

   where ω_{jm} are the mixing coefficients, μ_{jm} and Σ_{jm} are means the covariances matrices, respectively.

3. Finally, the initial state distribution (initial probability) is denoted as

$$\pi_i = p(q_1 = i), \qquad i = 1, \dots, h \tag{4.12}$$

In compact notation, an HMM can be represented by $\Lambda = (\pi, \mathbf{A}, \omega_{jm}, \mu_{jm}, \Sigma_{jm})$.

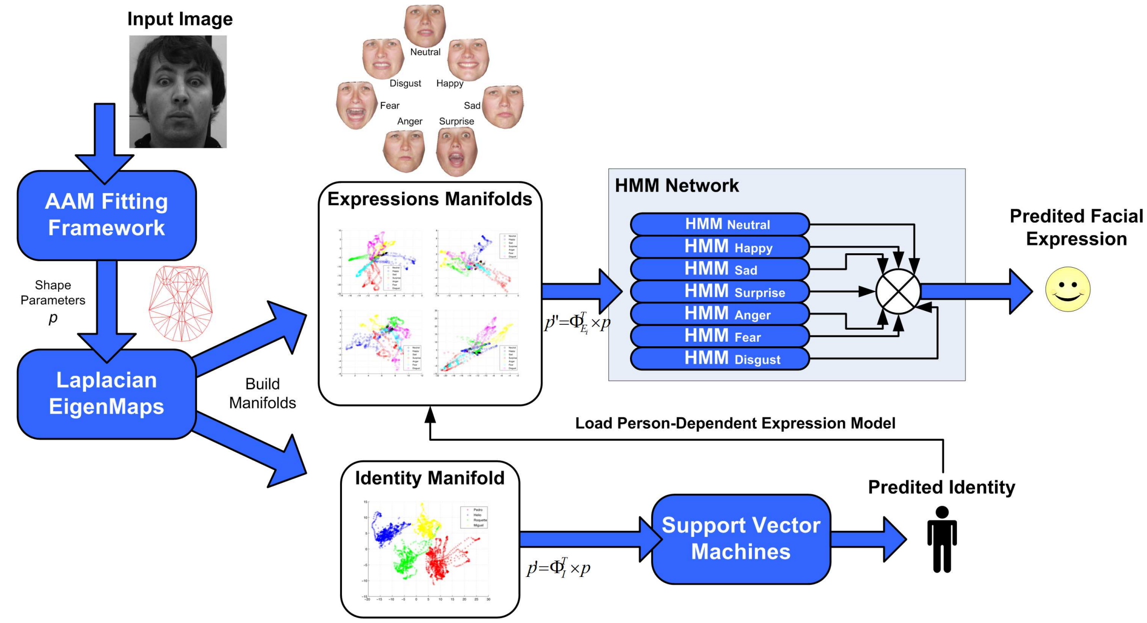


Figure 4.4: Overview of the proposed solution of recognize identity and facial expression. For an input image, the AAM extracts the shape parameters, **p**. These parameters are projected into a first manifold and the identity is predicted using SVM. The facial expression recognition mechanism is person-specific based. It uses a network of seven HMMs specialized in each facial emotion. The predicted expression is the one whose HMM sequence generated the highest probability.

4.5 The Recognition Approach

The proposed solution models both identity and facial expression in independent low dimensional manifolds, building person-specific expression models. The different manifolds were derived from embedding image data into a low dimensional subspace using Laplacian EigenMaps (LE) [51]. Learning these manifolds requires a discriminative facial representation from images, that is provided by the AAM fitting framework, see figure 4.3, where face images are represented by a set of sparse 2D feature point. As discriminatory features, instead of vectors with (x, y) feature points, there were used AAM related geometric features, i.e. regarding eq.4.1 the shape parameters, **p**, provide the same geometric information but using less dimensional features ($n << 2v$). Face normalization is done by only selecting the shape parameters that model deformation

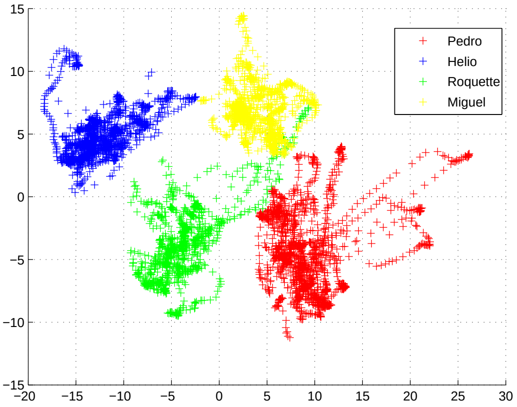


Figure 4.5: 2D representation of the identity manifold, $\Phi_{\mathbf{I}}$, learnt with AAM geometric related features for 4 persons.

(i.e. ignoring the 4 similarity parameters).

Both identity, $\Phi_{\mathbf{I}}$, and person-specific expression manifolds $\Phi_{\mathbf{E}_i}$ (with respect to subject i) were then learnt in a facial geometric feature space, consequently, an image sequence from a test subject describing a facial emotion is represented as a trajectory in the learnt manifold. Each image sequence starts with the subject at the neutral expression, then it exhibits an emotion into a maximum of expressivity and returns back to the neutral stage. See figure 4.6. These manifolds were build using LE representations for the shape parameters (which are related to face geometry). This approach maps the shape parameters, $\mathbf{p}$, into a less dimensional space, i.e. $\mathbf{p}' = \Phi_{\mathbf{I}}^T \mathbf{p}$, where the mapped features in our experiments acquire a huge discrimination power. As mentioned, two types of LE manifolds were derived: **(1)** The first type of manifold (lets call it identity manifold $\Phi_{\mathbf{I}}$) was built using data from all individuals, see figure 4.5; **(2)** the second type (the expression manifold $\Phi_{\mathbf{E}_i}$) uses data only from a single individual, emphasizing the differences in individual facial motion of the different expressions, see figure 4.6. This system holds an identity manifold and expression manifold for each of the individuals in the training set.

For recognition proposes, an approach with two stage cascade of classifiers was used. The first stage deals with identity recognition (across expression changes) where

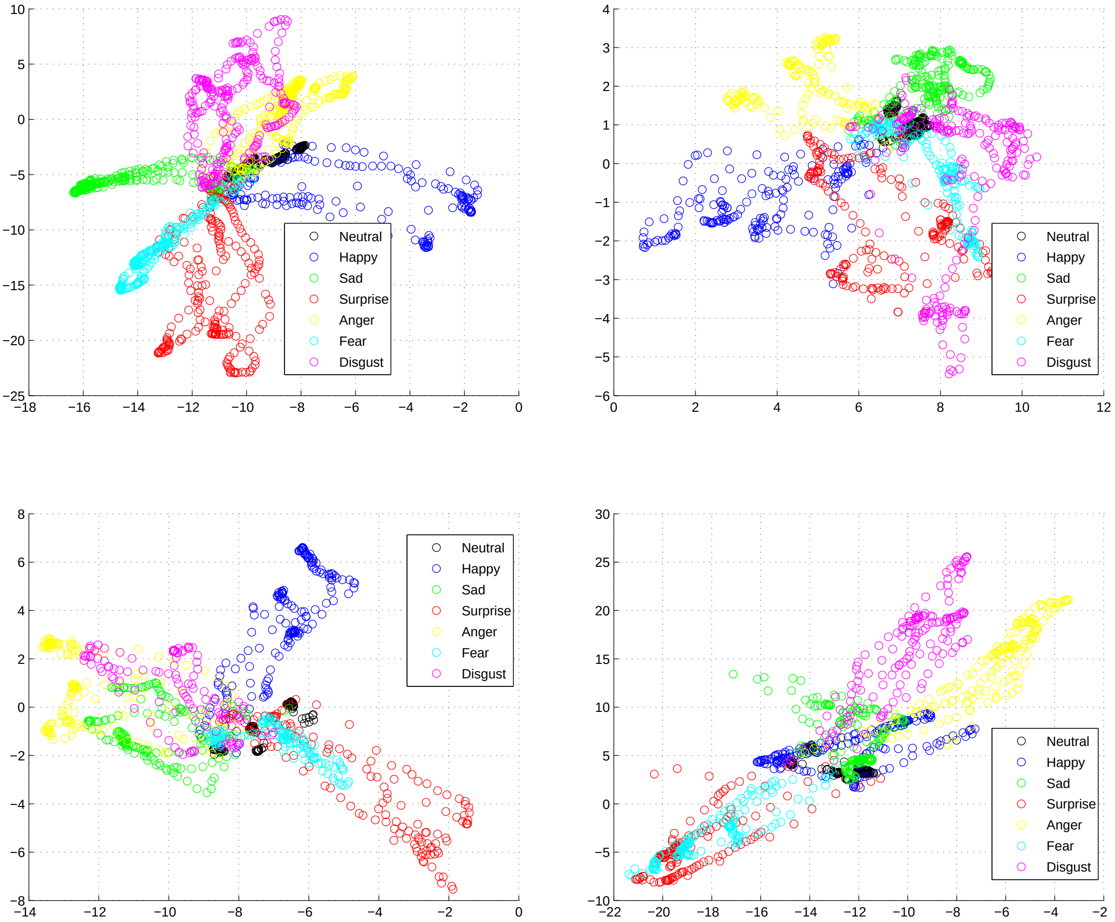


Figure 4.6: Low dimensional manifolds learnt with geometric AAM related features for 4 persons exhibiting 7 expressions several turns each. Top-right, top-left, bottom-right and bottom-right figures represent the expression models $\Phi_{\mathbf{E}_i}$ for person $i = 1, 2, 3$ and 4, respectively. Each facial expression sequence represents a trajectory in this space. All sequences start and finish at the neutral expression, hence, the high concentration of projected points over the neutral cluster.

a multiclass Support Vector Machines (SVM) [106][14] was trained with the identity manifold resulted data. The person-specific expression recognition, due the temporal dependency during the evolution of a facial emotion, is performed on the second stage using Hidden Markov Models (HMM) [48]. Seven HMM displaced in a parallel architecture were trained, each one specialized on the analyzed expressions. Input observation sequences (expression manifold projected features) fed each one of the HMM (shape parameters as latent variables) and the final decision was based on the sequence that yielded the highest (forward-backward) probability.

Figure 4.4 shows an overview of the proposed solution. Summarizing, it has a feature extracting mechanism and a two stage cascade classifiers trained with embedded manifold data. For an input image, the AAM fitting framework extracts the normalized shape parameters, $\mathbf{p}$. These parameters are projected into the identity manifold $\mathbf{p}' = \Phi_{\mathbf{I}}^T \mathbf{p}$, and the first SVM stage predict the identity i for the projected parameters $\mathbf{p}'$. The second stage loads the expression manifold, $\Phi_{\mathbf{E}_i}$, for the predicted identity. This stage consists on a network of seven HMMs displaced in a parallel architecture. The input features are projected into the expression manifold, $\mathbf{p}'' = \Phi_{\mathbf{E}_i}^T \mathbf{p}$ and the predicted expression is the one whose HMM model generated sequence yielded the highest probability.

4.6 Experimental Results

For the purpose of this work, a Facial Dynamics Database was built. It consists of several individuals, showing the seven basic facial expressions [62], namely: neutral expression, happiness, sadness, surprise, anger, fear and disgust. All facial emotions sequences were taken by starting and ending on the neutral expression. Each individual repeated all facial emotions four times. The dataset is formed by a total of 6770 images (640×480) taken from four individuals.

The AAM model was build using a total of 28 images (7 images for each of the 4 person). Since the AAM will be used to fit every frame of the captured database, it should held as much shape variation as possible. The training images were then composed by the most expressive images of the 7 emotions (from a random repetition sequence). These training images were hand annotated using $v = 58$ landmarks. Training the model holding 95% of shape and appearance variance produces an AAM with $n = 18$ shape parameters and $m = 29$ EigenFaces. All the 6770 frames of the Facial Dynamics Database were then fitted using the AAM model, retrieving the shape parameters, $\mathbf{p}$, for each frame.

Two main schemes were used for the manifold building: setting data for identity and setting the data for the expressions of each individual. A total of five manifolds were constructed (one identity manifold plus four individual-specific expression manifolds). These LE manifolds were build with both the number of adjacency graph neighbors, and the number of dimensions where the input features were projected into, found by cross-validation. Figures 4.5 and 4.6 show the manifolds produced for the identity and expressions, respectively. Regarding figure 4.6 it is noticed that person 1 (figure 4.6-top-right) is the most expressive and all facial emotions start and end from the neutral expression. This explains the high concentration of projected points over the neutral cluster.

On the first stage, a multiclass SVM [14] was trained with the input features of the identity manifold. The SVM classification was achieved using one-against-all voting scheme with a Gaussian Radial Basis Function (RBF) kernel. The kernel parameters and the missclassification penalty, were also found by cross-validation. Each individual-specific expression models in the second stage is composed by a network of seven HMM models displaced in a parallel architecture. These HMM models are specialized in each of the seven expressions. Representing h as the of number hidden states from a given HMM, h Gaussian Mixtures were fitted on the low dimensional data of the respective expression using K-means as the initial estimate. *Maximum likelihood* estimates of the parameters $(\pi, \mathbf{A}, \omega_{jm}, \mu_{jm}, \Sigma_{jm})$ were found using the Expectation-Maximization algorithm. The optimal number of states h and mixtures M were found by cross-validation analyzing the likelihood outputs on the re-estimation process (after the EM).

The final decision of the HMM network is made by evaluating the highest forward-backward probability on the sequence path provided by the Viterbi algorithm from all

	Indv1	**Indv2**	**Indv3**	**Indv4**
Indv1	**98.11**	0.09	1.79	0
Indv2	1.32	**98.67**	0	0
Indv3	2.93	0.29	**94.50**	2.27
Indv4	1.29	0.13	2.32	**96.25**

Overall recognition rate = 96.88%

Table 4.1: Identity model confusion matrix.

of the seven HMM.

4.6.1 Performance Evaluation

To evaluate the performance of the proposed solution, the dataset was divided into 4 fold for cross validation **F1**, **F2**, **F3** and **F4**, that matches to the four repetitions of all expressions that each subject has made. The results shown are of the form of confusion matrices that were obtained from the cross-validation of the four folds (i.e. [test **F1**, train **F2**, **F3**, **F4**] [test **F2**, train **F1**,**F3**,**F4**] [test **F3**, train **F1**,**F2**,**F4**] and finally [test **F4**, train **F1**, **F2**, **F3**]).

Identity and expression models were evaluated independently. Table 4.1 displays the confusion matrix for the identity recognition and table 4.2 shows the confusion matrices for the expression models for each person in the dataset. Baseline comparative results taken by a static based recognition [67] (still image classification using a multiclass one-against-all SVM with a Radial Basis Function kernel) are shown in table 4.3. Notice that, due the HMM based recognition, the results in table 4.2 could be misleading. This table shows classification for each of the 6770 frames, but when the observations don't have length enough the HMM don't produce reliable results, misclassifying many frames (that happens during the start of an emotion when no previous information is available). For this reason, the HMM network decision at the end of each observation sequence (full expression) is also shown at table 4.2).

Figures 4.7 and 4.8 shows several examples of the overall proposed approach. It is shown the projection into the identity manifold and the trajectory described on the respective expression manifolds.

Person 1	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**	Full Sequence						
Neut	**58.40**	0	3.05	0	16.03	0	22.52	**3**	0	0	0	1	0	0
Happ	1.25	**95.00**	0	0	3.75	0	0	0	**4**	0	0	0	0	0
Sad	0.59	0	**97.92**	1.47	0	0	0	0	0	**4**	0	0	0	0
Surp	0	0	0.66	**99.34**	0	0	0	0	0	0	**4**	0	0	0
Ang	0	2.69	5.09	0.59	**87.72**	1.20	2.69	0	0	0	0	**4**	0	0
Fear	0	0	0	29.37	2.23	**68.40**	0	0	0	0	1	0	**3**	0
Disg	0	0	4.14	0.95	32.80	0	**62.10**	0	0	0	0	1	0	**3**

Overall recognition rate = 81.27% 25/28 sequences

Person 2	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**	Full Sequence						
Neut	**47.32**	1.67	0	0	0	41.94	9.06	**2**	0	0	0	0	2	0
Happ	0	**70.34**	0	25.85	0	3.80	0	0	**3**	0	1	0	0	0
Sad	0	0.35	**93.43**	0.69	0	1.38	4.15	0	0	**4**	0	0	0	0
Surp	0	0	0.38	**97.32**	0.76	1.53	0	0	0	0	**4**	0	0	0
Ang	0	0	1.84	0	**91.70**	0.92	5.53	0	0	0	0	**4**	0	0
Fear	0	1.23	2.05	31.14	0	**61.47**	4.09	0	0	0	2	0	**2**	0
Disg	15.30	0.65	0	9.12	1.95	2.93	**70.03**	1	0	0	0	0	0	**3**

Overall recognition rate = 75.95% 22/28 sequences

Person 3	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**	Full Sequence						
Neut	**61.30**	20.10	0	0	18.59	0	0	**2**	1	0	0	1	0	0
Happ	0.86	**96.53**	0	2.59	0	0	0	0	**4**	0	0	0	0	0
Sad	0.44	0	**94.76**	0.87	2.18	1.31	0.44	0	0	**4**	0	0	0	0
Surp	23.40	0	0	**76.06**	0	0.53	0	1	0	0	**3**	0	0	0
Ang	0	2.38	2.38	1.90	**92.86**	0.48	0	0	0	0	0	**4**	0	0
Fear	21.25	0	6.87	16.87	1.25	**46.25**	7.50	1	0	0	1	0	**2**	0
Disg	27.03	1.35	16.21	0	9.46	1.35	**44.60**	1	0	1	0	0	0	**2**

Overall recognition rate = 73.20% 21/28 sequences

Person 4	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**	Full Sequence						
Neut	**25.00**	8.00	0	39.00	28.00	0	0	**1**	0	0	2	1	0	0
Happ	0	**95.79**	0	0.47	0	0	3.74	0	**4**	0	0	0	0	0
Sad	1.51	30.65	**53.26**	6.53	8.04	0	0	0	1	**3**	0	0	0	0
Surp	1.40	2.80	0	**66.35**	2.80	26.63	0	0	0	0	**3**	0	1	0
Ang	0	0.87	2.19	4.82	**87.72**	0.43	3.95	0	0	0	0	**4**	0	0
Fear	0	0	0	22.00	0	**80.00**	0	0	0	0	1	0	**3**	0
Disg	0	0	0	6.47	0	0.49	**93.03**	0	0	0	0	0	0	**4**

Overall recognition rate = 71.30% 22/28 sequences

Table 4.2: Confusion matrices of the facial expression recognition experiments using the HMM. Each table represents the individual results. The information in the right, labeled as Full Sequence, shows the recognition results by using the HMM with full sequence observations. Note that, a total of 6770 images were used in this evaluation.

Person 1	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**
Neut	**69.85**	9.16	2.29	0	0.76	1.14	16.79
Happ	0	**84.58**	3.33	10.41	1.66	0	0
Sad	0	0	**100**	0	0	0	0
Surp	0.66	0	0	**99.33**	0	0	0
Ang	2.40	0	0.89	0.59	**84.43**	0.29	11.37
Fear	0	0.74	0	38.66	0	**60.59**	0
Disg	2.54	0	0	37.57	20.70	0	**39.17**

Overall recognition rate = 76.85%

Person 2	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**
Neut	**67.78**	0	6.37	0	0	25.83	0
Happ	1.14	**78.70**	0	17.11	0	3.04	0
Sad	1.73	0	**86.85**	5.53	0	0	5.88
Surp	0.76	25.95	0.76	**41.60**	0	26.71	4.19
Ang	1.38	0	1.84	0	**79.26**	0.46	17.05
Fear	2.86	0	2.04	57.37	0	**33.61**	4.09
Disg	1.62	17.26	3.58	22.80	1.62	2.93	**50.16**

Overall recognition rate = 62.56%

Person 3	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**
Neut	**43.71**	0	20.10	25.62	0	10.55	0
Happ	3.89	**80.52**	0.43	6.49	0	3.89	4.76
Sad	8.29	0	**72.48**	0	10.48	2.62	6.11
Surp	5.31	6.91	0	**65.95**	0	21.80	0
Ang	4.28	0.47	25.71	0	**61.90**	0.95	6.66
Fear	21.25	23.12	0	18.75	0	**23.13**	13.75
Disg	10.13	2.02	37.16	10.81	5.40	2.02	**32.43**

Overall recognition rate = 54.30%

Person 4	**Neut**	**Happ**	**Sad**	**Surp**	**Ang**	**Fear**	**Disg**
Neut	**52.50**	17.50	0	18.00	0	0	12.00
Happ	4.67	**90.19**	0	3.73	0	1.14	0
Sad	2.01	12.56	**42.71**	0	0	0	42.71
Surp	1.86	2.80	0	**56.54**	0	32.71	6.07
Ang	2.19	0	0	0	**55.70**	0	42.10
Fear	1.43	3.34	0	16.26	0	**75.60**	3.34
Disg	0.49	6.46	0	0	0.99	0	**92.03**

Overall recognition rate = 66.47%

Table 4.3: Confusion matrices of the facial expression recognition experiments taken using the SVM (still images - static based recognition). Here presented as baseline results. Again, each table represents the individual results. In the same way, a total of 6770 images were used in this evaluation.

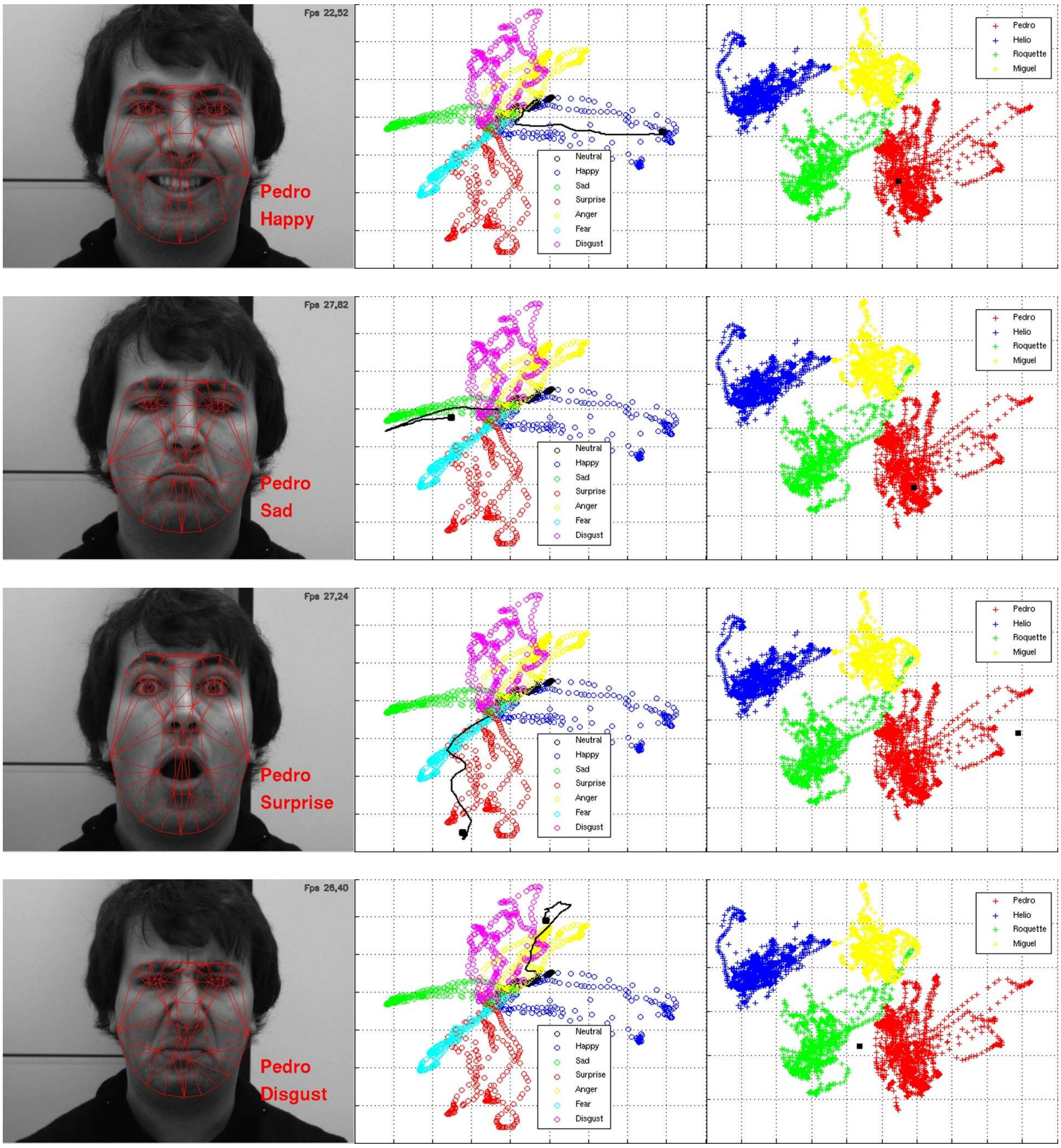


Figure 4.7: Examples of identity and expression recognition. The left images show the AAM fitting, the expression trajectory on the manifold is represented as a black path at center image and the projected test point into the identity manifold is the black dot at right image. The full video sequence can be seen at http://www.isr.uc.pt/~pedromartins/Videos/PhD.

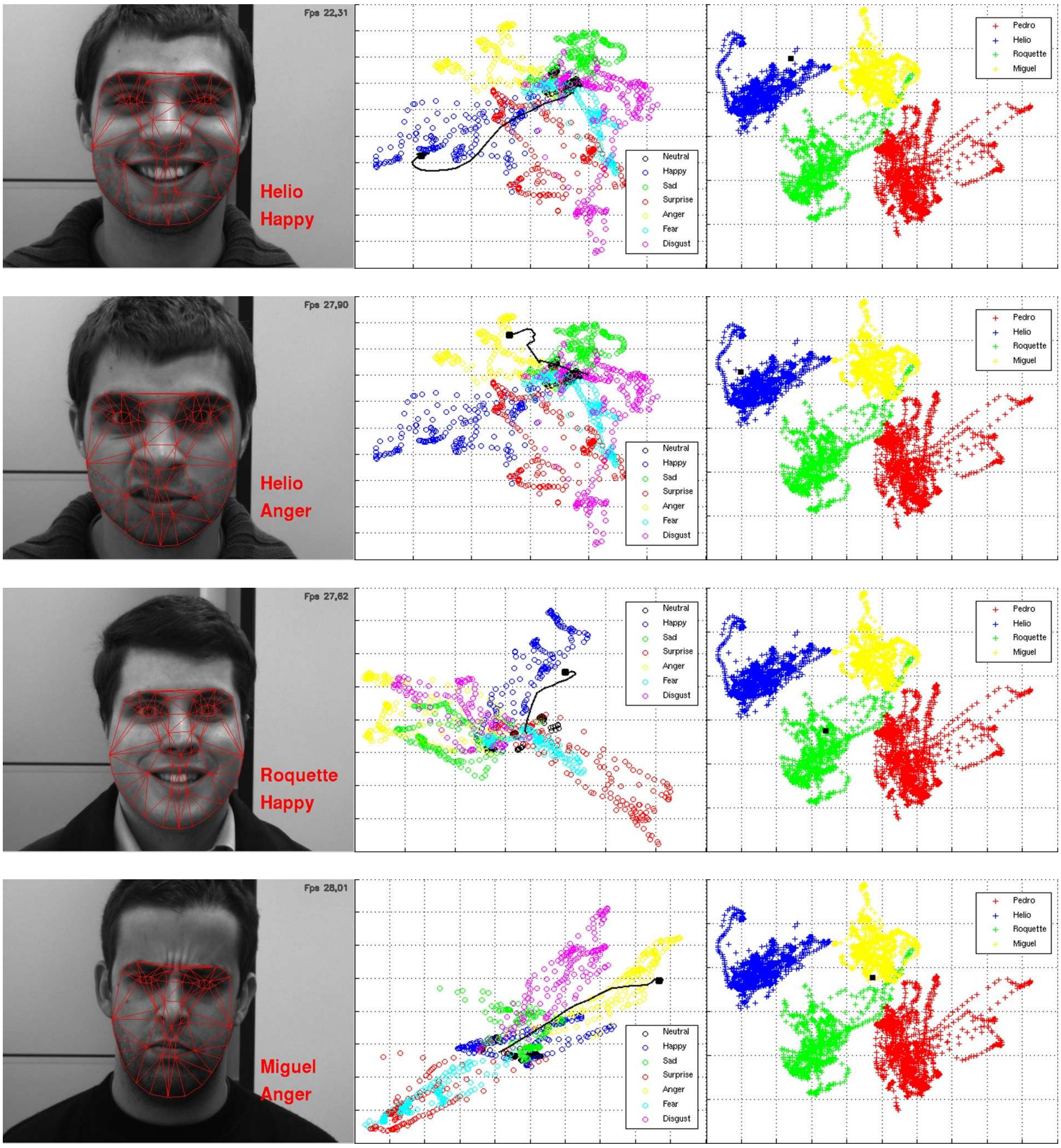


Figure 4.8: Additional examples of identity and expression recognition. The left images show the AAM fitting, the expression trajectory on the manifold is represented as a black path at center image and the projected test point into the identity manifold is the black dot at right image.

4.7 Conclusions

Human identity and facial expression recognition were achieved using a two stage classifier approach using low dimensional representation of the geometry of the face. Facial geometry related features were extracted using the Active Appearance Models (AAM) and low dimensional manifolds for identity and person-specific expression were derived using Laplacian EigenMaps (LE). For an input image, the AAM fitting framework extracts the normalized shape parameters and the first SVM stage predicts the identity for the projected parameters. The second stage is composed by a network of seven Hidden Markov Models (HMM), each one specialized on the several facial emotions analyzed. The normalized shape parameters are projected into the expression manifold of the predicted individual and the predicted expression is the one whose HMM generated sequence yielded the highest probability. For evaluation proposes a database was build having 6770 images captured from four people exhibiting seven different emotions. Our four fold cross-validation results show that the system is able to recognize an overall 96.8% in the identity. Since it was used person-specific expression models, the facial expression is dependent of each individual. In our dataset the most expressive individual achieves an overall recognition rate of 81.2% and the less expressive 71.3%.

Chapter 5

Conclusion

Advanced issues on facial and identity recognition requires basic face handling, namely non-rigid face registration. Model-based approaches are effective methods that can be used to overcome the large variation in shape and texture that images of human faces can present.

This thesis presents its main contributions in the face alignment/registration domain. Two different approaches where studied: generative and discriminative alignment methods (chapter 2 and 3, respectively). In short, the generative 2D Active Appearance Model (AAM) was extended to deal with a full perspective projection model and two new fitting algorithms, their efficient versions and robust to outliers extensions were proposed. The contributions in discriminative techniques include two novel global optimization strategies (DBASM and BASM). Both make inference in a MAP sense and align images using second order statistics of the latent variables. The difference is that the BASM formulation extends the DBASM by continuously update the prior distribution, therefore delivering more accurate results.

Recognition tasks, such as identity and facial expression classification were also addressed (chapter 4). A smaller contribution is made in this recognition field. Still, an efficient and effective face and emotion recognition system was presented. The overall approach uses a low dimensional representation of the face geometry and a network of Hidden Markov Models (HMMs). The system determines the facial expression from the extracted facial motion across time (facial dynamics).

More detailed descriptions are presented below.

Generative Face Alignment

A novel enhanced version of the 2D Active Appearance Models [38] (AAM) is proposed in this thesis. The 2.5D AAM, presented in chapter 2, extends the standard AAM to work with a full perspective projection model. The 2.5D AAM combines a 3D Point Distribution Model (PDM) and a 2D appearance model whose control points are defined by perspective projections of the PDM. The full six Degrees of Freedom (6 DOF) of the face are modeled by continuously integrate small pose changes at each frame since the beginning of tracking.

Two main fitting algorithms, the Simultaneous Forwards Additive (SFA), the Normalization Forwards Additive (NFA) and their computationally efficient approximations are proposed (ESFA and ENFA). Robust SFA and NFA solutions (RSFA and RNFA, respectively), taking into account partial and self occlusions are also proposed. All efficient versions (ESFA, ENFA, ERSFA and ERNFA) have shown a substantial improvement in the fitting performance, being more robust to noise and able to converge from far initial estimates, requiring less computational effort. The 2.5D AAM when compared with the 2D AAM or the 2D+3D AAM versions, has shown to better handle unseen data, converge faster and presented higher fitting success rates from far initial estimates.

The 2.5D AAM can achieve a high fitting accuracy, but it has the cost of labeling training examples. For best performance, the appearance model must be able to generated a valid template, i.e. the target individual must be included in the model building process (the main drawback of all the generative techniques).

Discriminative Face Alignment

This class of methods, presented in chapter 3, were designed to overcome the generalization limitation of the generative/holistic approaches (fitting in unseen data). Discriminative methods make use of an ensemble of local feature detectors whose locations are constrained by a PDM. Typically, fitting such model involves a two step approach: a local search (obtaining response maps for each landmark) and a global optimization strategy that finds the shape parameters that jointly maximize all the detection responses.

A Discriminative Bayesian Active Shape Model (DBASM) is proposed with a new global optimization strategy that efficiently solves the global alignment. DBASM infers both the shape and the pose parameters, in a maximum a posteriori (MAP) sense, by means of a Linear Dynamical System (LDS). This approach maintains 2$^{\text{nd}}$ order statistics of the shape and pose parameters, which represents the confidence in the current parameters estimate. A second Bayesian global optimization strategy, Bayesian Active Shape Models (BASM), an extension of DBASM is also described. BASM was designed to explicitly model the prior distribution by means of recursive Bayesian estimation. The prior distribution of the data was modeled as being Gaussian. The mean and covariance were assumed to be unknown and are treated as random variables.

A comparison between several face parts descriptors is also included, showing that the MOSSE filters [28] produce correlation filters that are notably stable, being particularly well-suited to the task of generic face alignment. Several DBASM and BASM evaluations were performed in unseen data using several image datasets, including the challenging Labeled Faces in the Wild (LFW). These discriminative techniques shown to significantly outperform other state-of-the-art fitting solutions.

Identity and Facial Expression Recognition

Finally, the chapter 4 proposes a two step identity and facial expression recognition approach that relies on a low dimensional representation of the geometry of the face. Face geometry is extracted from input images using the AAM and low dimensional manifolds were then derived using Laplacian EigenMaps (LE). The first stage uses a Support Vector Machines (SVM) to establish identity across expression changes. The second stage deals with person-specific facial expression recognition and is composed by a network of several Hidden Markov Models (HMM), each one specialized in a given facial emotion. The decision was made by the sequence that yielded the highest probability. The results in our database (6770 images captured from four people exhibiting the six basic emotions plus the neutral) show that the system is able to successful recognize the identity with 96.8%. Since it was used person-specific expression models, the facial expression is dependent of each individual. The most expressive individual achieves an overall recognition rate of 81.2% and the less expressive 71.3%.

5.1 Future Work

5.1.1 Unconstrained Non-Rigid Image Registration

The discriminative performance of DBASM/BASM [71][72] could be increased by enhancing the following components:

1. **Extend the likelihood term to a non-parametric distribution:** In [71] the global optimization was treated as an Bayesian inference problem, where the posterior distribution of the PDM parameters (and 2D pose) is inferred in a MAP sense. Even the response maps are being nonparametrically approximated (using the mean-shift algorithm), where only a single Gaussian (for each landmark) is extracted and used as likelihood term. The inferred posterior distribution is in fact a Gaussian as both likelihood and prior are also Gaussians. The likelihood term can be extend to be a non-parametric distribution (our at least a finite mixture of Gaussians). This can be done by using nonparametric Bayesian inference techniques [89] such as Markov Chain Monte Carlo (MCMC) methods.

2. **Shape representation - use a non-parametric shape model:** Like other previous works, DBASM uses a linear shape model (the PDM) that is built by applying a PCA on a representative dataset (typical with a few hundreds of images). Significant improvements are expected by relaxing the linear assumption to a non-parametric representation (which supplies the neglected PCA missing components).

3. **Deal with 3D head pose - extend to non-parametric 3D shape data:** Comparing 2D with 3D image alignment methods, the first is indeed more computational efficient, although the latter clearly have the potential to deliver more

accurate solutions, in particular for extreme head poses. According to the previous point a 3D non-parametric shape model is desirable.

4. **Use 3D depth information:** Recently, low cost 3D depth capable cameras (e.g. Kinect) had become very popular. An interesting addition is to seamlessly fuse the 3D depth information (multiple likelihoods) in the current formulation to increase the fitting performance.

5.1.2 Pose-Invariant Facial Dynamics Recognition

Developing robust and generalizable models for identity and facial expression recognition requires to establish a common reference frame, also known as the canonical referential. Considering that 2D face images are projections of 3D faces, the projective shape-space, which holds information about the spatial configuration of the landmarks that are invariant to the camera perspective, is of most importance in expression analysis. Since the projective shape-space is not well defined in the mathematical community, projective shapes in constrained situations can be approximated with affine shapes. Affine shape-space for facial landmark configurations has Grassmannian properties [5] and therefore nonrigid facial deformations due to various expressions can be represented as points on the Grassmann manifold (also a sequence can be modeled as a trajectory in this manifold). To model the sequential shape-data it is mandatory to take into account the Riemannian structure of the space in order to extract all the underlying information.

The goal is to extend one of the most common used sequential data classification algorithm (the Hidden Markov Models) to this shape-space (Grassmann manifold) by flattening using **Rolling Mapping:** Inference problems on Riemannian manifolds are usually addressed by embedding the manifolds into Euclidean spaces. In this paradigm, the embedding is obtained by fattening the manifold via local diffeomorphisms (tangent spaces). However, flattening the manifold through tangent spaces is not free of drawbacks. The exponential map is onto but only one-to-one in a neighborhood of a point (only locally defined). Therefore, the inverse mapping (logarithmic map) is uniquely valid only around a small neighborhood of that point. The idea is to project all the data from the manifold onto its affine tangent space at a particular point and then perform the classification there. To eliminate the distortion induced by local diffeomorphisms, the mathematical concept called Rolling Map [44] can be used. The novelty in this approach is that the manifold will be firstly rolled (without slipping or twisting) as a rigid body, then the given data is unwrapped onto the affine tangent space (having Euclidean properties) where the classification can be performed.

Bibliography

[1] *Bayesian Data Analysis*. Chapman & Hall/CRC, 2nd edition, 2004.

[2] A.Andreopoulos and J.K.Tsotsos. A novel algorithm for fitting 3-d active appearance models: Applications to cardiac mri segmentation. In *Scandinavian Conference on Image Analysis*, 2005.

[3] A.Andreopoulos and J.K.Tsotsos. Efficient and generalizable statistical models of shape and appearance for analysis of cardiac mri. *Medical Image Analysis*, 12(3):335–357, 2008.

[4] A.Bartoli. Groupwise geometric and photometric direct image registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2098–2108, December 2008.

[5] A.Edelman, T.A.Arias, and S.T.Smith. The geometry of algorithms with orthogonality constraints. *SIAM - Journal of Matrix Analysis and Applications*, 20(2):303–353, 1999.

[6] A.Sattar and R.Seguier. Mvaam (multi-view active appearance model) optimized by multi-objective genetic algorithm. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–8, 2008.

[7] A.Sattar, Y.Aidarous, and R.Seguier. Gagm-aam: A genetic optimization with gaussian mixtures for active appearance models. In *IEEE International Conference on Image Processing*, pages 3220–3223, 2008.

[8] A.Sattar, Y.Aidarous, S.Le.Gallou, and R.Seguier. Face alignment by 2.5d active appearance model optimized by simplex. In *International Conference on Computer Vision Systems (ICVS)*, March 2007.

[9] A.U.Batur and M.H.Hayes. Adaptive active appearance models. *IEEE Transactions on Image Processing*, 14(11):1707–1721, November 2005.

[10] B.Abboud and F.Davoine amd M.Dang. Facial expression recognition and synthesis based on an appearance model. *Signal Processing Image Communication*, 19(8):723–740, September 2004.

[11] B.Lucas and T.Kanade. An iterative image registration technique with an application to stereo vision (darpa). In *DARPA Image Understanding Workshop*, pages 121–130, April 1981.

[12] O. Bottema. *Topics in Elementary Geometry*. Springer, 2008.

[13] C. Butakoff and A.F. Frangi. Multi-view face segmentation using fusion of statistical shape and appearance models. *Computer Vision and Image Understanding*, 114(3):311–321, March 2010.

[14] C-C.Chang and C.-J.Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[15] C.Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society, Series B*, 53(2):285–339, 1991.

[16] C.Hu, J.Xiao, I.Matthews, S.Baker, J.Cohn, and T.Kanade. Fitting a single active appearance model simultaneously to multiple images. In *British Machine Vision Conference*, September 2004.

[17] C.M.Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[18] C.Shan, S.Gong, and P.McOwan. Dynamic facial expression recognition using a bayesian temporal manifold model. In *British Machine Vision Conference*, pages 297–306, 2006.

[19] C.Shen, M.J.Brooks, and A.Hengel. Fast global kernel density mode seeking: Applications to localization and tracking. *IEEE Transactions On Image Processing*, 16(5):1457–1469, May 2007.

[20] C.W.Chen and C.C.Wang. 3d active appearance model for aligning faces in 2d images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS*, September 2008.

[21] T. Dalgleish and M. Power. *Handbook of Cognition and Emotion.* John Wiley & Sons Ltd, June 1999.

[22] D.Cristinacce and T.F.Cootes. Facial feature detection using adaboost with shape constraints. In *British Machine Vision Conference*, 2003.

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

[24] D.Cristinacce and T.F.Cootes. Boosted regression active shape models. In *British Machine Vision Conference*, 2007.

[25] D.Cristinacce and T.F.Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[26] D. DeMenthon and L.S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, June 1995.

[27] D.Pizarro, J.Peyras, and A.Bartoli. Light-invariant fitting of active appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

[28] D.S.Bolme, J.R.Beveridge, B.A.Draper, and Y.M.Lui. Visual object tracking using adaptive correlation filters. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[29] F.Dornaika and F.Davoine. View- and texture-independent facial expression recognition in videos using dynamic programming. In *IEEE International Conference on Image Processing*, 2005.

[30] F.Dornaika and J.Ahlberg. Fast and reliable active appearance model search for 3d face tracking. In *International Conference on Model-based Imaging, Rendering, Image Analysis and Graphical Special Effects*, pages 113–122, March 2003.

[31] F.Dornaika and J.Ahlberg. Fitting 3d face models for tracking and active appearance model training. *Image and Vision Computing*, 24:1010–1024, September 2006.

[32] FGNet. Talking face video, 2004.

[33] G.B.Huang, M.Ramesh, T.Berg, and E.L.-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[34] G.Hager and P.Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–39, October 1998.

[35] H.-S.Lee and D.Kim. Expression-invariant face recognition by facial expression transformations. *Pattern Recognition Letters*, pages 1797–1805, October 2008.

[36] I.Akhter, Y.Sheikh, S.Khan, and T.Kanade. Nonrigid structure from motion in trajectory space. In *Neural Information Processing Systems*, 2008.

[37] I.Matthews, J.Xiao, and S.Baker. 2d vs. 3d deformable face models: Representational power, construction, and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, October 2007.

[38] I.Matthews and S.Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(1):135–164, November 2004.

[39] I.M.Scott, T.F.Cootes, and C.J.Taylor. Improving appearance model matching using local image structure. In *Information Processing in Medical Imaging*, pages 258–269, 2003.

[40] J.Ahlberg. Candide-3 - an updated parameterized face. Technical Report LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linköping University, Sweden, 2001.

[41] J.Saragih, S.Lucey, and J.Cohn. Face alignment through subspace constrained mean-shifts. In *IEEE International Conference on Computer Vision*, 2009.

[42] J.Xiao, J.Chai, and T.Kanade. A closed-form solution to non-rigid shape and motion recovery. In *European Conference on Computer Vision*, May 2004.

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

[44] K.Hüper and F.S.Leite. On the geometry of rolling and interpolation curves on sn, son, and grassmann manifolds. *JDCS - Journal of Dynamical and Control Systems*, 13(4):467–502, 2007.

[45] K.Messer, J.Matas, J.Kittler, J.Luettin, and G.Maitre. XM2VTSDB: The extended M2VTS database. In *AVBPA*, 1999.

[46] L.Gu and T.Kanade. A generative shape regularization model for robust face alignment. In *European Conference on Computer Vision*, 2008.

[47] L.Liang, R.Xiao, F.Wen, and J.Sun. Face alignment via component based discriminative search. In *European Conference on Computer Vision*, 2008.

[48] L.R.Rabiner. A tutorial on hidden markov models and selected applications in speech processing. *Proceedings of IEEE*, 77(2):257–286, February 1989.

[49] L.Torresani, A.Hertzmann, and C.Bregler. Learning non-rigid 3d shape from 2d motion. In *Neural Information Processing Systems*, 2003.

[50] L.Yin, X.Chen, Y.Sun, T.Worm, and M.Reale. A high-resolution 3d dynamic facial expression database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 17–19, September 2008.

[51] M.Belkin and P.Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[52] M.B.Stegmann and D.Pedersen. Bi-temporal 3d active appearance models with applications to unsupervised ejection fraction estimation. In *International Symposium on Medical Imaging (SPIE)*, volume 5747, pages 336–350, February 2005.

[53] M.B.Stegmann and R.Larsen. Multi-band modelling of appearance. In *International Workshop on Generative Model-Based Vision*, 2002.

[54] M.G.Roberts, T.F.Cootes, and J.E.Adams. Robust active appearance models with iteratively rescaled kernels. In *British Machine Vision Conference*, volume 1, pages 302–311, 2007.

[55] M.Kass, A.Witkin, and D.Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[56] M.Kirby and L.Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, January 1990.

[57] M.Nordstrom, M.Larsen, J.Sierakowski, and M.Stegmann. The IMM face database - an annotated dataset of 240 face images. Technical report, Technical University of Denmark, DTU, May 2004.

[58] M.S.Bartlett, G.Littlewort, M.Frank, C.Lainscsek, I.Fasel, and J.Movellan. Fully automatic facial action recognition in spontaneous behavior. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 223–230, 2006.

[59] M.Turk and A.Pentland. Face recognition using eigenfaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.

[60] M.Zhou, Y.Wang, X.Feng, and Xiaoyan Wang. A robust texture preprocessing for aam. In *International Conference on Computer Science and Software Engineering*, 2008.

[61] O.Jesorsky, K.Kirchberg, and R.Frischholz. Robust face detection using the hausdorff distance. In *AVBPA*, 2001.

[62] P.Ekman, W.V.Friesen, and J.C.Hager. *The Facial Action Coding System*. Weidenfeld & Nicolson, 2nd edition, 2002.

[63] P.Felzenszwalb and D.P.Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, January 2005.

[64] P.Lucey, S.Lucey, M.Cox, S.Sridharan, and J.F.Cohn. Comparing object alignment algorithms with appearance variation: Forward-additive vs inverse-composition. In *IEEE International Workshop on Multimedia Signal Processing - MMSP*, pages 337–342, 2008.

[65] P.Martins. Active appearance models for facial expression recognition and monocular head pose estimation. Master's thesis, FCTUC - Faculty of Sciences and Technology, University of Coimbra, 2008.

[66] P.Martins and J.Batista. Identity and expression recognition on low dimensional manifolds. In *IEEE International Conference on Image Processing*, 2009.

[67] P.Martins and J.Batista. Simultaneous identity and expression recognition using face geometry on low dimensional manifolds. In *Iberian Conference on Pattern Recognition and Image Analysis*, 2009.

[68] P.Martins and J.Batista. Towards generic fitting using discriminative active appearance models embedded on a riemannian manifold. In *International Conference on Computer Vision Theory and Applications*, 2010.

[69] P.Martins and J.Batista. Towards generic fitting using multiple features discriminative active appearance models. In *IEEE International Conference on Image Processing*, 2010.

[70] P.Martins, R.Caseiro, and J.Batista. Face alignment through 2.5d active appearance models. In *British Machine Vision Conference*, 2010.

[71] P.Martins, R.Caseiro, J.F.Henriques, and J.Batista. Discriminative bayesian active shape models. In *European Conference on Computer Vision*, 2012.

[72] P.Martins, R.Caseiro, J.F.Henriques, and J.Batista. Let the shape speak - discriminative face alignment using conjugate priors. In *British Machine Vision Conference*, 2012.

[73] P.Mittrapiyanuruk, G.N.DeSouza, and A.C.Kak. Accurate 3d tracking of rigid objects with occlusion using active appearance models. In *IEEE Workshop on Moition and Video Computing*, pages 90–95, 2005.

[74] P.Tresadern, H.Bhaskar, S.Adeshina, C.Taylor, and T.F.Cootes. Combining local and global shape models for deformable object matching. In *British Machine Vision Conference*, 2009.

[75] P.Viola and M.Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, July 2002.

[76] P.Yang, Q.Liu, X.Cui, and D.Metaxas. Facial expression recognition using encoded dynamic features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[77] R.-E.Fan, K.-W.Chang, C.-J.Hsieh, X.-R.Wang, and C.-J.Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, (9):1871–1874, 2008.

[78] Vincent Rabaud. Vincent's Structure from Motion Toolbox. `http://vision.ucsd.edu/~vrabaud/toolbox/`.

[79] R.Gross, I.Matthews, and S.Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23(1):1080–1093, November 2005.

[80] R.Gross, I.Matthews, and S.Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, 2006.

[81] R.Larsen, M.B. Stegmann, S.Darkner, S.Forchhammer, T.F.Cootes, and B.K.Ersbøll. Texture enhanced appearance models. *Computer Vision and Image Understanding*, 106(1):20–30, April 2007.

[82] R.Navarathna, S.Sridharan, and S.Lucey. Fourier active appearance models. In *IEEE International Conference on Computer Vision*, 2011.

[83] S.Baker and I.Matthews. Equivalence and efficiency of image alignment algoritms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1090–1097, December 2001.

[84] S.Baker and I.Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(1):221–255, March 2004.

[85] S.Baker, R.Gross, and I.Matthews. Lucas kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, CMU Robotics Institute, November 2003.

[86] S.Baker, R.Patil, K.M.Cheung, and I.Matthews. Lucas kanade 20 years on: A unifying framework: Part 5. Technical Report CMU-RI-TR-04-64, CMU Robotics Institute, November 2004.

[87] S.C.Mitchell, J.G.Bosch, B.P.F.Lelieveldt, R.J.van der Geest, J.H.C.Reiber, and M.Sonka. 3-d active appearance models: Segmentation of cardiac mr and ultrasound images. *IEEE Transactions on Medical Imaging*, 21(9):1167–1178, September 2002.

[88] S.E.Ayala-Raggi, L.Altamirano-Robles, and J.Cruz-Enriquez. Automatic face interpretation using fast 3d illumination-based aam models. *Computer Vision and Image Understanding*, 115(2):194–210, February 2011.

[89] S.Ghosal and A.W. van der Vaart. *Theory of Nonparametric Bayesian Inference.* Cambridge University Press, 2009.

[90] S.Romdhani and T.Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *IEEE International Conference on Computer Vision*, pages 59–66, 2003.

[91] S.Sclaroff and J.Isidoro. Active blobs: Region based deformable appearance models. *Computer Vision and Image Understanding*, 89(3):197–225, 2003.

[92] S.Zambal. 3d active appearance models for segmentation of cardiac mri data. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, August 2005.

[93] T.F.Cootes and C.J.Taylor. Active shape models - smart snakes. In *British Machine Vision Conference*, 1992.

[94] T.F.Cootes and C.J.Taylor. Constrained active appearance models. In *IEEE International Conference on Computer Vision*, 2001.

[95] T.F.Cootes and C.J.Taylor. Statistical models of appearance for computer vision. Technical report, Imaging Science and Biomedical Engineering, University of Manchester, 2004.

[96] T.F.Cootes and C.J.Taylor. An algorithm for tuning an active appearance model to new data. In *British Machine Vision Conference*, 2006.

[97] T.F.Cootes, C.J.Taylor, D.H.Cooper, and J.Graham. Training models of shape from sets of examples. In *British Machine Vision Conference*, 1992.

[98] T.F.Cootes, C.J.Taylor, D.H.Cooper, and J.Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[99] T.F.Cootes, G.J.Edwards, and C.J.Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[100] T.F.Cootes, G.V.Wheeler, K.N.Walker, and C.J.Taylor. View-based active appearance models. *Image and Vision Computing*, 20:657–664, 2002.

[101] B.J. Theobald, I.Matthews, and S.Baker. Evaluating error functions for robust active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 149–154, April 2006.

[102] T.Ojala, M.Pietikinen, and T.Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[103] T.Xiang, M.K.H.Leung, and S.Y.Cho. Expression recognition using fuzzy spatio-temporal modeling. *Pattern Recognition*, 41(1):204–216, January 2008.

[104] U.Paquet. Convexity and bayesian constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[105] V.Blanz and T.Vetter. A morphable model for the synthesis of 3d faces. In *Comput Graph (ACM) SIGGRAPH*, pages 187–194, 1999.

[106] V.Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

[107] X.Hou, S.Z.Li, H.Zhang, and Q.Cheng. Direct appearance models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[108] X.Liu. Generic face alignment using boosted appearance model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[109] X.Liu. Discriminative face alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1941–1954, November 2009.

[110] X.Liu. Video-based face model fitting using adaptive active appearance model. *Image and Vision Computing*, 28(7):1162–1172, 2010.

[111] X.Liu, F.W.Wheeler, and P.H.Tu. Improved face model fitting on video sequences. In *British Machine Vision Conference*, September 2007.

[112] Y.Cheon and D.Kim. Natural facial expression recognition using differential-aam and manifold learning. *Pattern Recognition*, 42(7):1340–1350, July 2009.

[113] Y.Du and X.Lin. Emotional facial expression model building. *Pattern Recognition Letters*, 24(16):2923–2934, December 2003.

[114] Y.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, February 2001.

[115] Y.Wang, S.Lucey, and J.Cohn. Enforcing convexity for improved alignment with constrained local models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[116] Y.Zhou, L.Gu, and H.J.Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via bayesian inference. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

Appendix A

2.5D AAM

A.1 The Jacobian of The Warp Partial Differentials

Defining the elements $m_{0_{ij}}$ of the base projection matrix $\mathbf{M}_0$ as

$$\underbrace{\begin{bmatrix} m_{0_{11}} & m_{0_{12}} & m_{0_{13}} & m_{0_{14}} \\ m_{0_{21}} & m_{0_{22}} & m_{0_{23}} & m_{0_{24}} \\ m_{0_{31}} & m_{0_{32}} & m_{0_{33}} & m_{0_{34}} \end{bmatrix}}_{\mathbf{M}_0} = \mathbf{K} \begin{bmatrix} \mathbf{R}_0 & \mathbf{t}_0 \end{bmatrix} \tag{A.1}$$

the quantities $\xi_1, \ldots, \xi_6$ and $\Xi_1, \ldots, \Xi_6$ are scalar values given by

$$\begin{aligned} \xi_1 &= m_{0_{11}} \phi_i^{x_k} + m_{0_{12}} \phi_i^{y_k} + m_{0_{13}} \phi_i^{z_k} & \xi_4 &= m_{0_{11}} \psi_j^{x_k} + m_{0_{12}} \psi_j^{y_k} + m_{0_{13}} \psi_j^{z_k} \\ \xi_2 &= m_{0_{21}} \phi_i^{x_k} + m_{0_{22}} \phi_i^{y_k} + m_{0_{23}} \phi_i^{z_k} & \xi_5 &= m_{0_{21}} \psi_j^{x_k} + m_{0_{22}} \psi_j^{y_k} + m_{0_{23}} \psi_j^{z_k} \\ \xi_3 &= m_{0_{31}} \phi_i^{x_k} + m_{0_{32}} \phi_i^{y_k} + m_{0_{33}} \phi_i^{z_k} & \xi_6 &= m_{0_{31}} \psi_j^{x_k} + m_{0_{32}} \psi_j^{y_k} + m_{0_{33}} \psi_j^{z_k} \end{aligned} \tag{A.2}$$

$$\begin{bmatrix} \Xi_1 \\ \Xi_2 \\ \Xi_3 \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{x_k} + p_i \phi_i^{x_k} + \sum_{j \neq i}^{n} p_j \phi_j^{x_k} + \sum_{j=1}^{6} q_j \psi_j^{x_k} + s_\psi^{x_k} \\ s_0^{y_k} + p_i \phi_i^{y_k} + \sum_{j \neq i}^{n} p_j \phi_j^{y_k} + \sum_{j=1}^{6} q_j \psi_j^{y_k} + s_\psi^{y_k} \\ s_0^{z_k} + p_i \phi_i^{z_k} + \sum_{j \neq i}^{n} p_j \phi_j^{z_k} + \sum_{j=1}^{6} q_j \psi_j^{z_k} + s_\psi^{z_k} \\ 1 \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} \Xi_4 \\ \Xi_5 \\ \Xi_6 \end{bmatrix} = \mathbf{M}_0 \begin{bmatrix} s_0^{x_k} + \sum_{i=1}^{n} p_i \phi_i^{x_k} + q_j \psi_j^{x_k} + \sum_{i \neq j}^{6} q_i \psi_i^{x_k} + s_\psi^{x_k} \\ s_0^{y_k} + \sum_{i=1}^{n} p_i \phi_i^{y_k} + q_j \psi_j^{y_k} + \sum_{i \neq j}^{6} q_i \psi_i^{y_k} + s_\psi^{y_k} \\ s_0^{z_k} + \sum_{i=1}^{n} p_i \phi_i^{z_k} + q_j \psi_j^{z_k} + \sum_{i \neq j}^{6} q_i \psi_i^{z_k} + s_\psi^{z_k} \\ 1 \end{bmatrix} . \qquad (A.4)$$

A.2 Details on the Efficient Fitting Algorithms

```
Precompute:
The 2.5D parametric models
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ (fig. 2.8)
Gradients of the template $\nabla A_0(\mathbf{x_p})$
Gradients of the Eigen images $\nabla A_i(\mathbf{x_p})$
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$
    Error image $\mathbf{E}(\mathbf{x_p})_{\text{sfa}}$ using eq.2.18
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute efficient SD images $\mathbf{SD}(\mathbf{x_p})_{\text{esfa}}$
    using eq.2.29
    Hessian matrix and its inverse
    $\mathbf{H}_{\text{esfa}} = \sum_{\mathbf{x_p}} \mathbf{SD}(\mathbf{x_p})_{\text{esfa}}^T \mathbf{SD}(\mathbf{x_p})_{\text{esfa}}$
    Parameters updates
    $\Delta\mathbf{r} = \mathbf{H}_{\text{esfa}}^{-1} \sum_{\mathbf{x_p}} \mathbf{SD}(\mathbf{x_p})_{\text{esfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{sfa}}$
    Update parameters $\mathbf{r} \leftarrow \mathbf{r} + \Delta\mathbf{r}$
    Update pose offset
    $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $||\Delta \boldsymbol{r}|| \leq \varepsilon$ or max. number of
iterations reached ;
```

Algorithm 10: Efficient SFA.

```
Precompute:
The 2.5D parametric models
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ (fig. 2.8)
Gradients of the template $\nabla A_0(\mathbf{x_p})$
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$
    Error image $\mathbf{E}(\mathbf{x_p})_{\text{lk}}$ using eq.2.23
    Estimate $\boldsymbol{\lambda}$ using eq.2.25
    Normalized Error image $\mathbf{E}(\mathbf{x_p})_{\text{nfa}}$
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute efficient SD images $\mathbf{SD}(\mathbf{x_p})_{\text{enfa}}$
    using eq.2.30
    Hessian matrix and its inverse
    $\mathbf{H}_{\text{enfa}} = \sum_{\mathbf{x_p}} \mathbf{SD}(\mathbf{x_p})_{\text{enfa}}^T \mathbf{SD}(\mathbf{x_p})_{\text{enfa}}$
    $\begin{bmatrix} \Delta\mathbf{p} \\ \Delta\mathbf{q} \end{bmatrix} = \mathbf{H}_{\text{enfa}}^{-1} \sum_{\mathbf{x_p}} \mathbf{SD}(\mathbf{x_p})_{\text{enfa}}^T \mathbf{E}(\mathbf{x_p})_{\text{efa}}$
    Update $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ and $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$
    Update pose offset
    $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $\left\| \begin{matrix} \Delta \boldsymbol{p} \\ \Delta \boldsymbol{q} \end{matrix} \right\| \leq \varepsilon$ or max. number of
iterations reached ;
```

Algorithm 11: Efficient NFA.

```
Precompute:
The 2.5D parametric models
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ (fig. 2.8)
Gradients of the template $\nabla A_0(\mathbf{x_p})$
Gradients of the Eigen images $\nabla A_i(\mathbf{x_p})$
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$
    Evaluate triangle visibility
    Error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{sfa}}$ using eq.2.18
    Estimate the weight mask $\rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{sfa}})$
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute efficient SD images $\mathbf{SD}(\mathbf{x_p})_{\mathrm{esfa}}$ using eq.2.29
    Weighted Hessian matrix $\mathbf{H}_{\mathrm{ersfa}} = \sum_{\mathbf{x_p}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{sfa}})\mathbf{SD}(\mathbf{x_p})^T_{\mathrm{esfa}}\mathbf{SD}(\mathbf{x_p})_{\mathrm{esfa}}$
    Parameters updates $\Delta\mathbf{r} = \mathbf{H}^{-1}_{\mathrm{ersfa}} \sum_{\mathbf{x_p}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{sfa}})\mathbf{SD}(\mathbf{x_p})^T_{\mathrm{esfa}}\mathbf{E}(\mathbf{x_p})_{\mathrm{sfa}}$
    Update parameters $\mathbf{r} \leftarrow \mathbf{r} + \Delta\mathbf{r}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $||\Delta \boldsymbol{r}|| \leq \varepsilon$ or max. number of iterations reached ;
```

Algorithm 12: Efficient Robust SFA.

```
Precompute:
The 2.5D parametric models
Evaluate $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{x}_k}$ and $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p})}{\partial \mathbf{y}_k}$ (fig. 2.8)
Gradients of the template $\nabla A_0(\mathbf{x_p})$
repeat
    Update pose reference $\Psi(s_\psi)$ with eq.2.6
    Warp input image $\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$
    Evaluate triangle visibility
    Error image $\mathbf{E}(\mathbf{x_p})_{\mathrm{rnfa}}$ using eq.2.37
    Estimate the weight mask $\rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{rnfa}})$
    Hessian Appearance $\mathbf{H}_{\mathrm{A}}$ with eq.2.40
    Update app. parameters $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \Delta\boldsymbol{\lambda}$
    Recompute $\mathbf{E}(\mathbf{x_p})_{\mathrm{rnfa}}$ using eq.2.37
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{p}}$ (eq.2.41)
    Find Jacobian $\frac{\partial \mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial \mathbf{q}}$ (eq.2.44)
    Compute efficient SD images $\mathbf{SD}(\mathbf{x_p})_{\mathrm{enfa}}$ using eq.2.30
    Weighted Hessian matrix $\mathbf{H}_{\mathrm{ernfa}} = \sum_{\mathbf{x_p}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{rnfa}})\mathbf{SD}(\mathbf{x_p})^T_{\mathrm{enfa}}\mathbf{SD}(\mathbf{x_p})_{\mathrm{enfa}}$
    $\begin{bmatrix}\Delta\mathbf{p}\\ \Delta\mathbf{q}\end{bmatrix} = \mathbf{H}^{-1}_{\mathrm{ernfa}} \sum_{\mathbf{x_p}} \rho(\mathbf{E}(\mathbf{x_p})^2_{\mathrm{rnfa}})\mathbf{SD}(\mathbf{x_p})^T_{\mathrm{enfa}}\mathbf{E}(\mathbf{x_p})_{\mathrm{rnefa}}$
    Update $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ and $\mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}$
    Update pose offset $s_\psi \leftarrow s_\psi + \sum_{j=1}^{6} \psi_j \Delta q_j$
until $\left\|\begin{matrix}\Delta p\\ \Delta q\end{matrix}\right\| \leq \varepsilon$ or max. number of iterations reached ;
```

Algorithm 13: Efficient Robust NFA.

A.3 Piecewise Affine Warp

The piecewise affine warp is composed by sets of affine warps between corresponding triangles of the mesh. The base triangles are found by partitioning the convex hull of the projected mean shape, $s_{0\mathbf{p}}$, using the Delaunay triangulation, and each pixel belonging to a given triangle is mapped to its correspondent triangle using barycentric coordinates.

As mentioned, two meshes are involved in the warping procedure: the projected base mesh $s_{0\mathbf{p}}$ (that is fixed) with the triangle vertexes $< (x^0_{p_i}, y^0_{p_i})^T$, $(x^0_{p_j}, y^0_{p_j})^T$, $(x^0_{p_k}, y^0_{p_k})^T >$ and the current projected mesh $s_{\mathbf{p}}$ with the triangles vertexes coordinates $< (x_{p_i}, y_{p_i})^T$, $(x_{p_j}, y_{p_j})^T$, $(x_{p_k}, y_{p_k})^T >$, being $(i, j, k = \#$ triangles). See figure 2.4. Both meshes depend on $\mathbf{p}$ and $\mathbf{q}$ by eqs.2.9 and 2.1.

The barycentric coordinates α, β, used in eq.2.9, are given by

$$\alpha = \frac{(x_p - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y_p - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})}{(x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})} \tag{A.5}$$

$$\beta = \frac{(y_p - y^0_{p_i})(x^0_{p_j} - x^0_{p_i}) - (x_p - x^0_{p_i})(y^0_{p_j} - y^0_{p_i})}{(x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i})}, \tag{A.6}$$

and the eqs. 2.9, A.5, A.6, 2.1 and 2.8 can be combined into a single per-triangle affine warp, as

$$\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}) = (a_1 + a_2 x_p + a_3 y_p, a_4 + a_5 x_p + a_6 y_p)^T \tag{A.7}$$

where a_1, a_2, a_3, a_4, a_5 and a_6 are the affine parameters that are given by

$$\begin{aligned}
a_1 &= (x_{p_i}(x^0_{p_j} y^0_{p_k} - y^0_{p_j} x^0_{p_k}) + x^0_{p_i}(x_{p_k} y^0_{p_j} - y^0_{p_k} x_{p_j}) + y^0_{p_i}(x^0_{p_k} x_{p_j} - x^0_{p_j} x_{p_k}))/\Delta \\
a_2 &= (y^0_{p_k}(x_{p_j} - x_{p_i}) + y^0_{p_i}(x_{p_k} - x_{p_j}) + y^0_{p_j}(x_{p_i} - x_{p_k}))/\Delta \\
a_3 &= (x^0_{p_k}(x_{p_i} - x_{p_j}) + x^0_{p_j}(x_{p_k} - x_{p_i}) + x^0_{p_i}(x_{p_j} - x_{p_k}))/\Delta \\
a_4 &= (y_{p_i}(x^0_{p_j} y^0_{p_k} - y^0_{p_j} x^0_{p_k}) + x^0_{p_i}(y_{p_k} y^0_{p_j} - y^0_{p_k} y_{p_j}) + y^0_{p_i}(x^0_{p_k} y_{p_j} - x^0_{p_j} y_{p_k}))/\Delta \\
a_5 &= (y^0_{p_k}(y_{p_j} - y_{p_i}) + y^0_{p_i}(y_{p_k} - y_{p_j}) + y^0_{p_j}(y_{p_i} - y_{p_k}))/\Delta \\
a_6 &= (x^0_{p_k}(y_{p_i} - y_{p_j}) + x^0_{p_j}(y_{p_k} - y_{p_i}) + x^0_{p_i}(y_{p_j} - y_{p_k}))/\Delta \\
&\text{with} \\
\Delta &= (x^0_{p_j} - x^0_{p_i})(y^0_{p_k} - y^0_{p_i}) - (y^0_{p_j} - y^0_{p_i})(x^0_{p_k} - x^0_{p_i}).
\end{aligned} \tag{A.8}$$

The affine parameters $a_1, \ldots, a_6$ need only to be computed once per triangle, not once per pixel. Also, and since the projected base mesh is fixed (i.e. there is always a constant warping frame), a lookup table that encodes the triangle identity speeds up the entire warping procedure. The algorithm 14 summarizes this section by showing the list of steps required to perform the piecewise affine warp.

```
Precompute: The triangle lookuptable
Evaluate the current mesh $s$ from $\mathbf{p}$ and $\mathbf{q}$ using eq.2.1
Find the full perspective mesh projection $s_{\mathbf{p}}$ with eq.2.8
Compute the affine parameters $(a_1, a_2, a_3, a_4, a_5, a_6)$ for each triangle using eqs.A.8
For each pixel $\mathbf{x_p}$ in the projected base mesh $s_{0_\mathbf{p}}$, lookup the triangle where $\mathbf{x_p}$ lies in
    and then lookup the corresponding values of $(a_1, \dots, a_6)$
Evaluate $\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})$ from eq.A.7 and bilinear interpolate to find $\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))$
```

Algorithm 14: Piecewise affine warp.

A.4 Simultaneous Forwards Additive Derivation

The nonlinear optimization at eq. 2.11:

$$\arg\min_{\mathbf{p},\mathbf{q},\boldsymbol{\lambda}} \sum_{\mathbf{x_p}\in s_{0\mathbf{p}}} \left[\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} \lambda_i \mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) \right]^2$$

can be solved by gradient descent using additive updates to the parameters as

$$\sum_{\mathbf{x}\in s_{0\mathbf{p}}} [\mathbf{A}_0(\mathbf{x_p}) + \sum_{i=1}^{m+2} (\lambda_i + \Delta\lambda_i)\mathbf{A}_i(\mathbf{x_p}) - \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p} + \Delta\mathbf{p}, \mathbf{q} + \Delta\mathbf{q}))]^2. \tag{A.9}$$

Using a first order Taylor expansion, the last term can be expressed as

$$\mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p} + \Delta\mathbf{p}, \mathbf{q} + \Delta\mathbf{q})) \approx \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})) + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}} \Delta\mathbf{p} + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{q}} \Delta\mathbf{q}, \tag{A.10}$$

and, the chain rule can be used on part of the second term of eq.A.10, giving

$$\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}} = \left[\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial x} \frac{\partial \mathbf{W}_x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}} + \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial y} \frac{\partial \mathbf{W}_y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}} \right]. \tag{A.11}$$

Rearranging the terms, results

$$\frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial \mathbf{p}} = \underbrace{\left[\begin{array}{cc} \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial x} & \frac{\partial \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))}{\partial y} \end{array} \right]}_{\nabla \mathbf{I}(\mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q}))} \underbrace{\left[\begin{array}{ccc} \frac{\partial \mathbf{W}x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_1} & \dots & \frac{\partial \mathbf{W}x(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_n} \\ \frac{\partial \mathbf{W}y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_1} & \dots & \frac{\partial \mathbf{W}y(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}_n} \end{array} \right]}_{\text{Jacobian of the Warp } \frac{\partial \mathbf{W}(\mathbf{x_p}, \mathbf{p}, \mathbf{q})}{\partial \mathbf{p}}}, \tag{A.12}$$

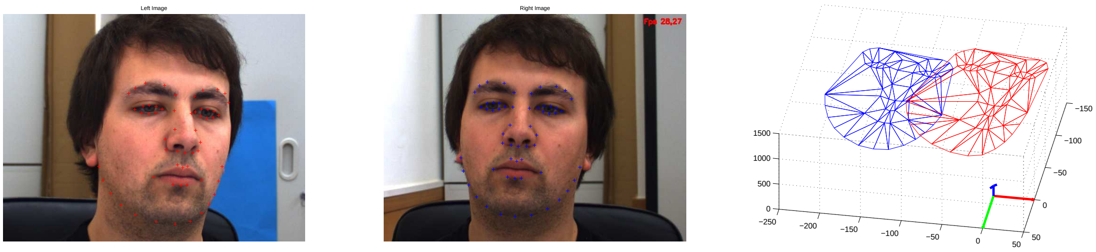


Figure A.1: Left and right images captured by a calibrated stereo system. Each shape annotation results from applying a 2D AAM. The 3D recovered structures (for each camera) are shown on the right picture. Red and blue colors respectively.

being $\nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))$ the gradients of the image $\mathbf{I}(\mathbf{x_p})$ evaluated at $\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})$ and the term $\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{p}}$ the Jacobian of the warp w.r.t. the shape parameters, $\mathbf{p}$.

Similarly for the pose parameters, $\mathbf{q}$, part of the last term of eq.A.10 can be written as

$$\frac{\partial\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))}{\partial\mathbf{q}} = \nabla\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))\frac{\partial\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q})}{\partial\mathbf{q}}. \tag{A.13}$$

Finally the eq.A.9, can be written as

$$\sum_{\mathbf{x_p}\in s_{0\mathbf{p}}}\left[\mathbf{A}_0(\mathbf{x_p})+\sum_{i=1}^{m+2}\lambda_i\mathbf{A}_i(\mathbf{x_p})+\sum_{i=1}^{m+2}\Delta\lambda_i\mathbf{A}_i(\mathbf{x_p})-\mathbf{I}(\mathbf{W}(\mathbf{x_p},\mathbf{p},\mathbf{q}))-\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-\nabla\mathbf{I}\frac{\partial\mathbf{W}}{\partial\mathbf{q}}\Delta\mathbf{q}\right]^2. \tag{A.14}$$

A.5 Building the 3D PDM From Stereo Data

The 3D PDM used in this work, was built using a fully calibrated stereo system where the 2D shape on each view was extracted by fitting a 2D AAM [38] using $v = 58$ landmarks. See figure A.1.

The classical triangulation algorithm was used to recover the 3D structure for each view. In short, the triangulation algorithm consists in finding the depths Z_l and Z_r from the normalized perspective projections $(x_l, y_l) = (\frac{X_l}{Z_l}, \frac{Y_l}{Z_l})$ and $(x_r, y_r) = (\frac{X_r}{Z_r}, \frac{Y_r}{Z_r})$ with (X_l, Y_l, Z_l) and (X_r, Y_r, Z_r) being the coordinates of the same 3D point in the left and right camera frame, all this, knowing the rotation $\mathbf{R}$ and translation $\mathbf{t}$ between cameras. The least-squares solution, using all the v points in each shape annotation, is given by

$$\begin{bmatrix} Z_{l_1} & \cdots & Z_{l_v} \\ Z_{r_1} & \cdots & Z_{r_v} \end{bmatrix} = \begin{bmatrix} -\mathbf{R} \begin{pmatrix} x_{r_1} & \cdots & x_{r_v} \\ y_{r_1} & \cdots & y_{r_v} \\ 1 & \cdots & 1 \end{pmatrix} & \begin{pmatrix} x_{l_1} & \cdots & x_{l_v} \\ y_{l_1} & \cdots & y_{l_v} \\ 1 & \cdots & 1 \end{pmatrix} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{t} & \cdots & \mathbf{t} \end{bmatrix}. \tag{A.15}$$

Using eqs.A.15, the 3D shape mesh samples from pairs of 2D image annotations can be retrieved, as illustrated in figure A.1. However, these mesh coordinates are expressed w.r.t. the camera coordinate frame and therefore the user head rotations are not correctly modeled. To overcome this problem, the PDM was converted into the base pose $(\mathbf{R}_0, \mathbf{t}_0)$ coordinate frame (as included in eq.2.8)[1], by firstly removing the mean from s_0, centering the mean shape around de origin[2] and then $\mathbf{R}_0$ and $\mathbf{t}_0$ were found by solving the following optimization problem:

$$\arg\min_{\theta,\gamma,t_z} \mathbf{K} \begin{bmatrix} \mathbf{R}_{pan}(\theta)\mathbf{R}_{roll}(\gamma) & \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix} \end{bmatrix} \begin{bmatrix} s_0^{x_1} \cdots s_0^{x_v} \\ s_0^{y_1} \cdots s_0^{y_v} \\ s_0^{z_1} \cdots s_0^{z_v} \\ 1 \cdots 1 \end{bmatrix} \tag{A.16}$$

where $\mathbf{R}_{pan}(\theta)$ and $\mathbf{R}_{roll}(\gamma)$ represent the pan and roll rotations matrices by θ and γ amount, respectively, that changes the 3D orientation of s_0. The t_z parameter is the translation along the camera optical axis from the centroid of the mean shape s_0.

The optimization in eq.A.16 is performed in four steps. First t_z is found by setting a desirable 2D mesh projection width over the image plane (p.e. 200 pixels) holding θ and γ equal to zero. This width value defines the base mesh projection size that is related to all the fitting algorithms computational complexity. The base mesh projection size define the constant warping frame described in the texture model section and

[1] Expressing the PDM w.r.t. another coordinate frame requires only changes on the rigid motion (s_0).

[2] It would be convenient to center s_0 around the neck axis, where the true head rotations are made. However, estimating the true neck coordinate frame is not in the scope of this work. We simply move the center of gravity of s_0 back and down 50mm as $s_0 \leftarrow (s_0^{x_i}, s_0^{y_i} - 50, s_0^{z_i} - 50), i = 1, \ldots, v$.

consequently the size of all the Steepest Descent images. Then θ and γ are optimized independently in order to hold a symmetric mesh projection. A symmetric shape is desirable to balance the model fitting, otherwise the AAM will perform better for user head rotations where the texture model holds more pixels.

Finally, the last step consist in optimize again for t_z using the previously found values of θ and γ, just to hold the desirable 2D mesh projection width. The base pose is then given by

$$\mathbf{R}_0 = \mathbf{R}_{pan}(\theta)\mathbf{R}_{roll}(\gamma) \text{ and } \mathbf{t}_0 = \begin{pmatrix} 0 \\ 0 \\ t_z \end{pmatrix}. \tag{A.17}$$

Appendix B

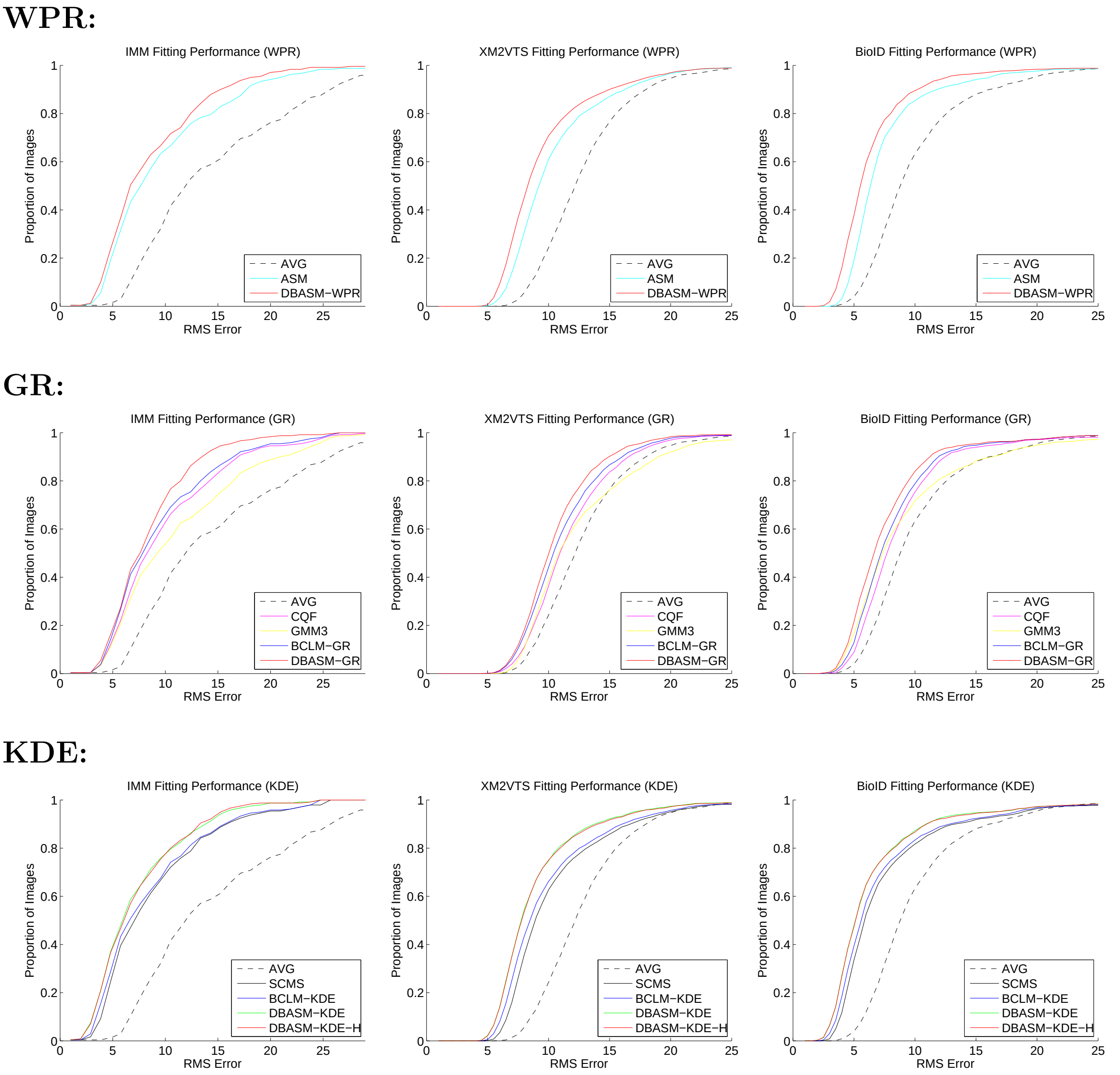
DBASM Evaluation

B.1 Additional Global Strategies Evaluation

Figures B.1 and B.2 show detailed fitting performance curves when using two different local detectors: the MOSSE filters and the linear SVM detectors, respectively. Fitting algorithms with comparable local strategies, namely Weighted Peak Responses (WPR), Gaussian Responses (GR) and Kernel Density Estimator (KDE) are shown individually. Similarly to section 3.4.3, the table B.1 shows quantitative values of figure B.2 (SVM local detectors) taken by setting a fixed RMS error amount (7.5 pixels). Each table entry show how many percentage of images converge with less or equal RMS error than the reference.

Figure B.3 shows tracking performance evaluation on the FGNET Talking Face [32] sequence using the linear SVM local detectors.

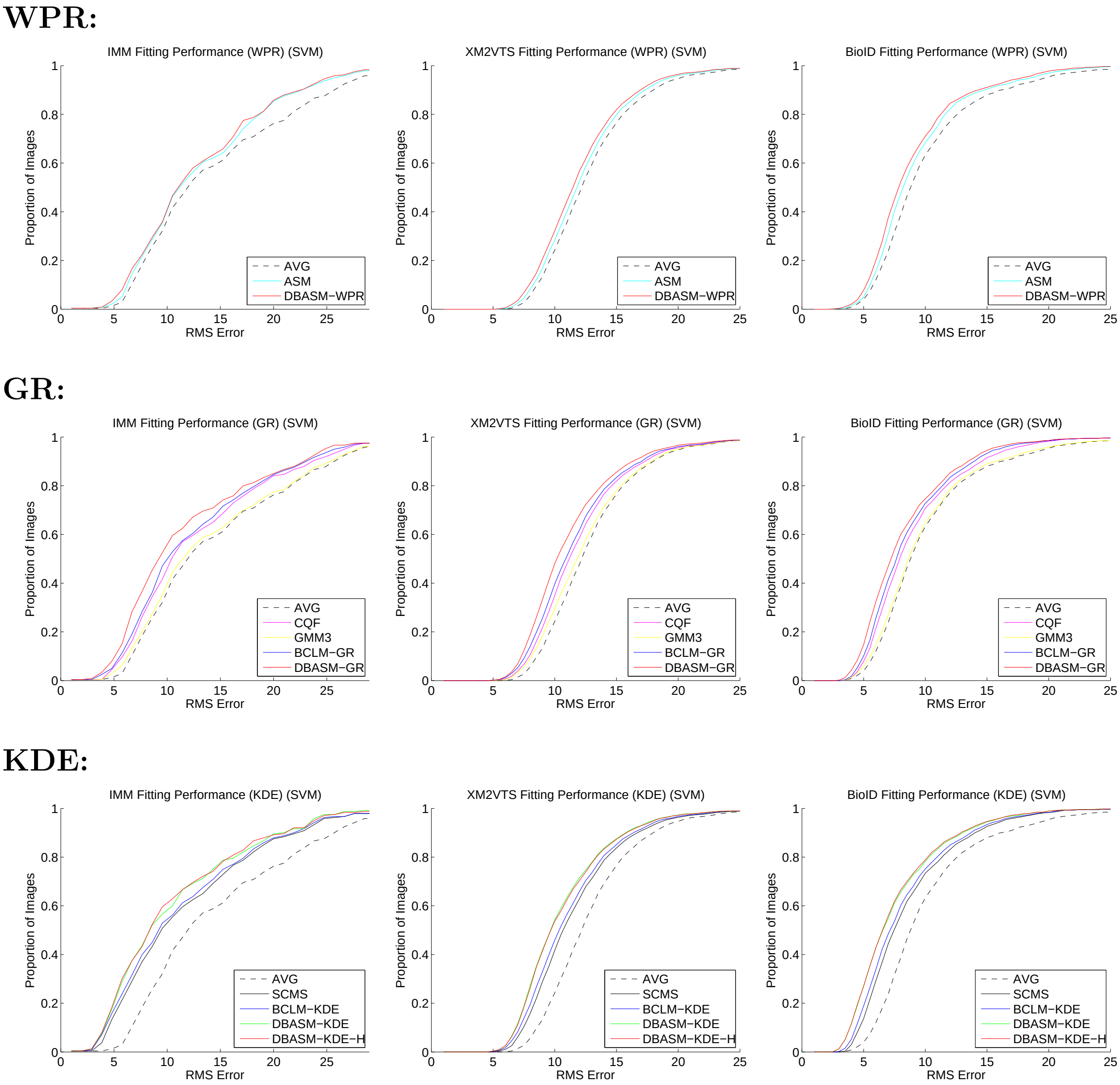
MOSSE Detectors



(a) IMM [57] database (b) XM2VTS [45] database (c) BioID [61] database

Figure B.1: Fitting performance curves when using MOSSE filter as local detector. Local strategies are shown by rows - Weighted Peak Responses (WPR), Gaussian Responses (GR) and Kernel Density Estimator (KDE). AVG means the initial estimate given by Adaboost [75] face detector. The results show that our proposed methods (DBASM-WPR, DBASM-GR, DBASM-KDE and DBASM-KDE-H) outperform all the others.

Linear SVM Detectors



(a) IMM [57] database (b) XM2VTS [45] database (c) BioID [61] database

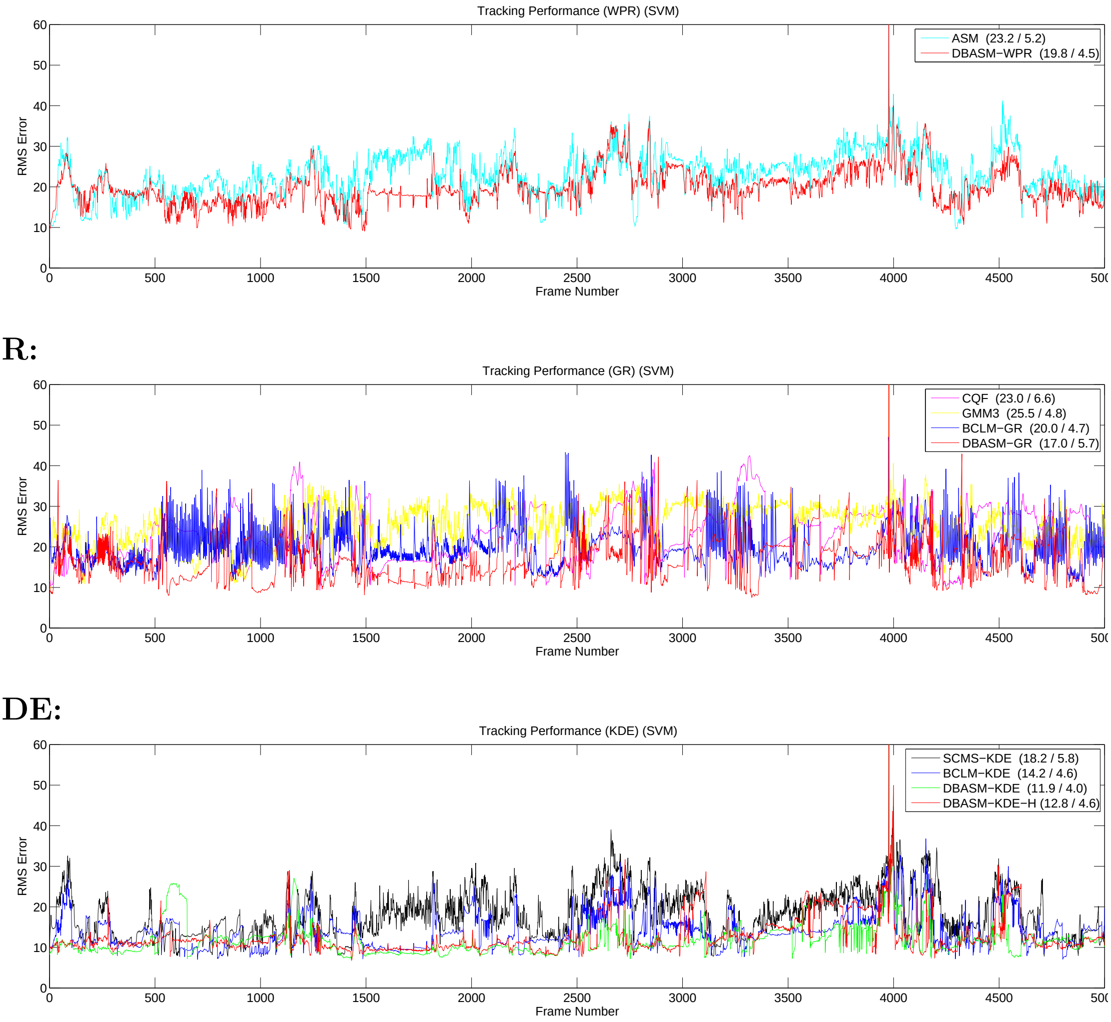
Figure B.2: Fitting performance curves when using the linear SVM detector. Local strategies are shown by rows - Weighted Peak Responses (WPR), Gaussian Responses (GR) and Kernel Density Estimator (KDE). The results show that our proposed methods (DBASM-WPR, DBASM-GR, DBASM-KDE and DBASM-KDE-H) outperform all the others.

Reference 7.5 RMS	IMM (240 images)		XM2VTS (2360 images)		BioID (1521 images)	
ASM	21.7		8.1		40.4	
DBASM-WPR (our method)	**22.5**	(+0.8)	**10.6**	(+2.5)	**45.1**	(+4.7)
CQF	26.2		10.6		43.9	
GMM3	20.0	(-6.2)	9.0	(-1.6)	34.0	(-9.9)
BCLM-GR	28.3	(+2.1)	14.0	(+3.4)	47.5	(+3.6)
DBASM-GR (our method)	**36.7**	(+10.5)	**18.8**	(+8.2)	**53.8**	(+9.9)
SCMS-KDE	37.1		15.7		50.4	
BCLM-KDE	40.0	(+2.9)	19.4	(+3.7)	53.6	(+3.2)
DBASM-KDE (our method)	**43.8**	(+6.7)	**27.3**	(+11.6)	61.2	(+10.8)
DBASM-KDE-H (our method)	43.3	(+6.2)	26.2	(+10.5)	**61.8**	(+11.4)

Table B.1: Quantitative results using the SVM linear detectors. The table shows quantitative values taken by setting a fixed RMS error amount (7.5 pixels - w.r.t. figure B.2). Each table entry show how many percentage of images converge with less or equal RMS error than the reference. Again, the results show that our proposed methods outperform all the other, using all the different local strategies WPR, GR and KDE.

Linear SVM Detectors

WPR:



GR:

KDE:

Figure B.3: Evaluation of the tracking performance of several fitting algorithms on the FGNET Talking Face [32] sequence using the linear SVM detectors. The values on legend box are the mean and standard deviation RMS errors, respectively. When using simpler detectors the results remain the same, DBASM methods are more stable and accurate.