uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Marius Daniel Cordea
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

Ph.D. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A 3D Anthropometric Muscle-based Active Appearance Model for Model-based Video Coding
TITRE DE LA THÈSE / TITLE OF THESIS

Emil Petriu
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Dorina Petriu
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. El Saddik

R. Hornsey (absent)

N. Georganas

R. Goubran

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A 3D Anthropometric Muscle-Based Active Appearance Model for Model-Based Video Coding

Marius Daniel Cordea,

Thesis submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. in Electrical and Computer Engineering

Ph.D. Thesis Advisors

Dr. Emil M. Petriu, University of Ottawa
Dr. Dorina C. Petriu, Carleton University

School of Information Technology and Engineering,
Ottawa-Carleton Institute of Electrical and Computer Engineering,
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-34123-0
Our file Notre référence
ISBN: 978-0-494-34123-0

Acknowledgements

I would like to take this opportunity to thank all the people who have been helpful to the completion of this thesis. First of all, I thank Dr. Emil M. Petriu and Dr. Dorina C. Petriu, my thesis supervisors, for their guidance and support during the course of this thesis. I would like to especially thank my family for the support and love they have given me my whole life.

Abstract

A 3D-Anthropometric-Muscle-Based Active Appearance Model for Model-Based Video Coding

The work of this thesis focuses in key areas of human-computer interaction (HCI), namely rigid facial motion recovery and facial expression analysis, and interpretation. Rigid motion recovery from image sequences is based on Structure-From-Motion (SFM) using Kalman Filter-based recursive algorithms. Facial expression analysis is performed by an Active Appearance Model (AAM), which is a statistical model, based on the estimation of linear models of shape and texture variation. The thesis integrates new developed algorithms into an Automatic Facial Tracking System (AFTS) for a low bit-rate videophone system.

The first contribution of this thesis is a new method for modeling the shape and appearance of three-dimensional (3D) human faces using a constrained 3D Active Appearance Model (AAM). The proposed algorithm is an extension of the classical 2D Active Appearance Model. It uses a generic 3D wireframe model of the face, based on two sets of controls: the anatomically motivated muscle actuators to model facial expressions and statistically based anthropometrical controls to model different facial types (*3D Anthropometric Muscle-Based Active Appearance Model* (3D AMB AAM). This allows describing a facial image in terms of a controlled model parameter set, hence providing both, a natural and a constrained basis for face segmentation and analysis. The generated face models are consequently simpler and less memory intensive compared to the classical appearance based models. The proposed method provides accurate fitting results by constraining solutions to be valid instances of a face model. Extensive image segmentation experiments demonstrate the accuracy of the proposed algorithm against the classical AAM.

The second contribution of this thesis is a new 3D tracking algorithm allowing real-time recovery of 3D position, orientation and facial expressions of a moving head. The described method uses a recursive motion estimation algorithm, namely an Extended Kalman Filter (EFK) to extract the head pose (global motion) and the newly developed 3D AMB AAM to extract the facial expressions (local motion). The resulting motion tracking system works in a realistic environment without makeup on the face, with an uncalibrated camera, and unknown lighting conditions and background. In order to validate the accuracy of the 3D head tracking system, a rapid calibration technique was developed using a sequence of images of a synthetic "standard" 3D head in lieu of a real head.

List of Terms

2D	Two Dimensional
2½D	Two and Half-Dimensional
3D	Three Dimensional
AAM	Active Appearance Model
AAU	Anthropometric Action Unit
ABS	Analysis by Synthesis
ACM	Active Contour Model
AE	Anthropometric-Expression
AFTS	Automatic Face Tracking System
AMB	Anthropometric Muscle-Based
AMB AAM	Anthropometric Muscle-Based Active Appearance Model
ASM	Active Shape Model
AU	Action Unit
bps	bits per second
BU	Boston University
CIA	Compositional Image Alignment
COP	Center of Projection
DAAM	Direct AAM
DOF	Degree-of-Freedom
EAU	Expression Action Unit
EKF	Extended Kalman Filter
EMFACS	Emotional FACS
FACS	Facial Action Coding System
FAP	Facial Animation Parameters
FG-NET	Face and Gesture Recognition Network
GPA	Generalized Procrustes Analysis
HCI	Human-Computer Interaction
HMM	Hidden Markov Models
KL	Karhunen-Loeve

KLT	Kanade-Lucas-Tomasi
KPCA	Kernel Principal Component Analysis
LDA	Linear Discriminant Analysis
MBVC	Model Based Video Coding
MDL	Minimum Description Length
MM	Morphable Model
MMI	Man Machine Interaction
MPEG	Moving Pictures Expert Group
NN	Neural Network
ORL	Olivetti Research Laboratory
PAL	Phase Alternating Line
PCA	Principal Component Analysis
PDA	Personal Digital Assistant
PDM	Point Distribution Model
POTS	Plain Old Telephone Service
PSNR	Peak Signal-to-Noise Ratio
RMSE	Root Mean Square Error
SFM	Structure-from-Motion
SNHC	Synthetic/Natural Hybrid Coding
SVM	Support Vector Machines

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

Nowadays many research activities are trying to model a natural human-computer interaction (HCI), based on daily human communication means such as body gestures, speech, and facial expressions. Verbal and visual communications are the main components of a HCI system. Speech recognition and synthesis are increasingly part of a large number of practical applications in verbal communication. Gesture and facial expression analysis, recognition and interpretation represent main issues in visual communication. The development of practical applications involving visual communication was a far-off goal in the past due to low computer power. Currently, the advances in computer technology make it possible to overcome the limitations in developing natural HCI systems. Such systems include automatic finding, tracking and modeling of human faces from image sequences. These represent important and challenging tasks in areas of image processing, pattern recognition and computer vision. There is an increased market for this technology. Besides HCI, direct applications include facial recognition, security, automotive safety, model-based coding for multimedia and telecommunication applications, virtual reality, medicine, education, and entertainment.

Facial recognition from still and video images is an emerging biometric research area with many commercial and law enforcement applications. Biometric solutions can be used in areas of border control, immigration applications, physical access control, computer and network security, restricted area surveillance, and to recognize people in zones of interest (banks, stores), or in databases (police, passport database etc.). Also, systems monitoring facial expression can preview adverse human behavior by signaling extreme emotions as early warnings.

Face and eye tracking help automakers monitor driver state for automotive safety. Moreover, knowing the head position of a passenger could be used to guide airbag deployment direction and strength.

Facial expression could play a significant role in an educational environment. For example, a face model that captures the expressions and anatomy of the human face may be used in tele-learning applications over low bit-rate channels.

Another application domain employing head pose and facial expressions is virtual reality. Specific applications include: interactive virtual worlds, games, and character animation. In a complex environment, computer generated characters capture the facial expressions and human emotions of a "live performer" without the use of sensors. Prerecorded scripts can control the synthesis of requested facial expressions.

Corrective plastic surgery and preoperative simulations in dental treatments need anatomically complete models of the person's head (with bone and soft tissue) [Koch96]. This way, surgeons have the capability to plan and practice complex operations. Recognizing facial movements can help physically impaired people to control devices. Also, computer-generated facial models may contribute to the rehabilitation of people with impairments of facial muscles, and represent a useful tool in psychology and nonverbal communication areas.

Multimedia applications range from desktop videoconferencing to interactive entertainment networks providing video-on-demand, information services, video games, and tele-shopping. The integration of motion video in multimedia environments is technologically one of the most demanding tasks, due to the high data rates and real-time constraints. In recent years, several video coding standards such as H.261, H.263, MPEG-1, and MPEG-2 have been introduced. These video coders utilize the statistics of the video signal without knowledge of the semantic content of the frames and can therefore be used for arbitrary scenes. In today's multimedia systems, an important issue is the efficient storage, transmission and manipulation of image sequences and 3D scenes. The new MPEG-4 Video and Synthetic/Natural Hybrid Coding (SNHC) standard [MPG02], [Per02], covers the emerging content-based image coding and image access as well as animation of 3D models. A priori knowledge of the scene contents can help representing objects by synthetic models. For example, the contents of a video telephony stream are known a priori that is sequences of head-and-shoulder images of a talking person. This semantic information can be used to encode the scene with a higher efficiency.

Encoding image objects with their synthetic models at the transmission end and decoding them at the receiver end has been a wide research area for more than three decades of *Model-Based Video Coding* (MBVC), also known as *Knowledge-Based Video Coding* [For83], [Aiz87], [Aiz89], [Wel91], [Pea95], [Hua01], [Eis00], [Ahl98], [Cor98], [Cor01], [Cor02], and [Str02]. The concept of MBVC of human faces was introduced in 1983 by Forchheimer and Fahlander [For83], and has emerged as a very low bit rate video compression method suitable for videoconferencing and videophone applications. Aizawa [Aiz87] and Welsh [Wel91] introduced realistic texture-mapping of the human face models.

Visual communications over low bandwidth (below 10 Kbps) channels is possible only if implementing high efficiency coding methods. One such candidate is the MBVC technique. The main advantages of MBVC are the low transmission rate needed for communication and low storage costs. MBVC can use existing communications channels such as POTS (plain old telephone service), wireless, and the Internet. The MBVC increases coding efficiency by using knowledge about the scene content and describing the real world geometry by 3D model objects (e.g. 3D face models). Additionally, object-specific functionalities like the animation of facial expressions should be provided. In a videoconferencing scenario, the principle of this compression is to generate parametric model of the image seen at the emission end, and to transmit only the characteristic parameters (pose, expressions, and texture updates) describing how the model changes in time. These change parameters are then used to animate the model of the image recovered at the reception end. Each image is analyzed and transformed into a 3D-wireframe model. Using a priori knowledge about the images to be transmitted, a wireframe model of a generic human face is molded on the live image of the particular person at the emission end. In order to represent faces by 3D models the sender has to firstly perform challenging tasks such as face detection, tracking and facial expression understanding. The 3D face modeling recovers color and texture of the person's face in the polygons of the wireframe, conveying a natural-look feeling. Fig. 1.1 presents the structural diagram of the one-way channel of the model-based videophone system [Cor98]. The emitting videophone sends the complete set of data on the model geometry, color and texture to the receiving end only once. After that, it needs to transmit only the description of changes in the image model (pose and expression), which can result in a data transmission rate as low as 1 Kbps [Gir94], [Eis00] without reducing the resolution of the original image.

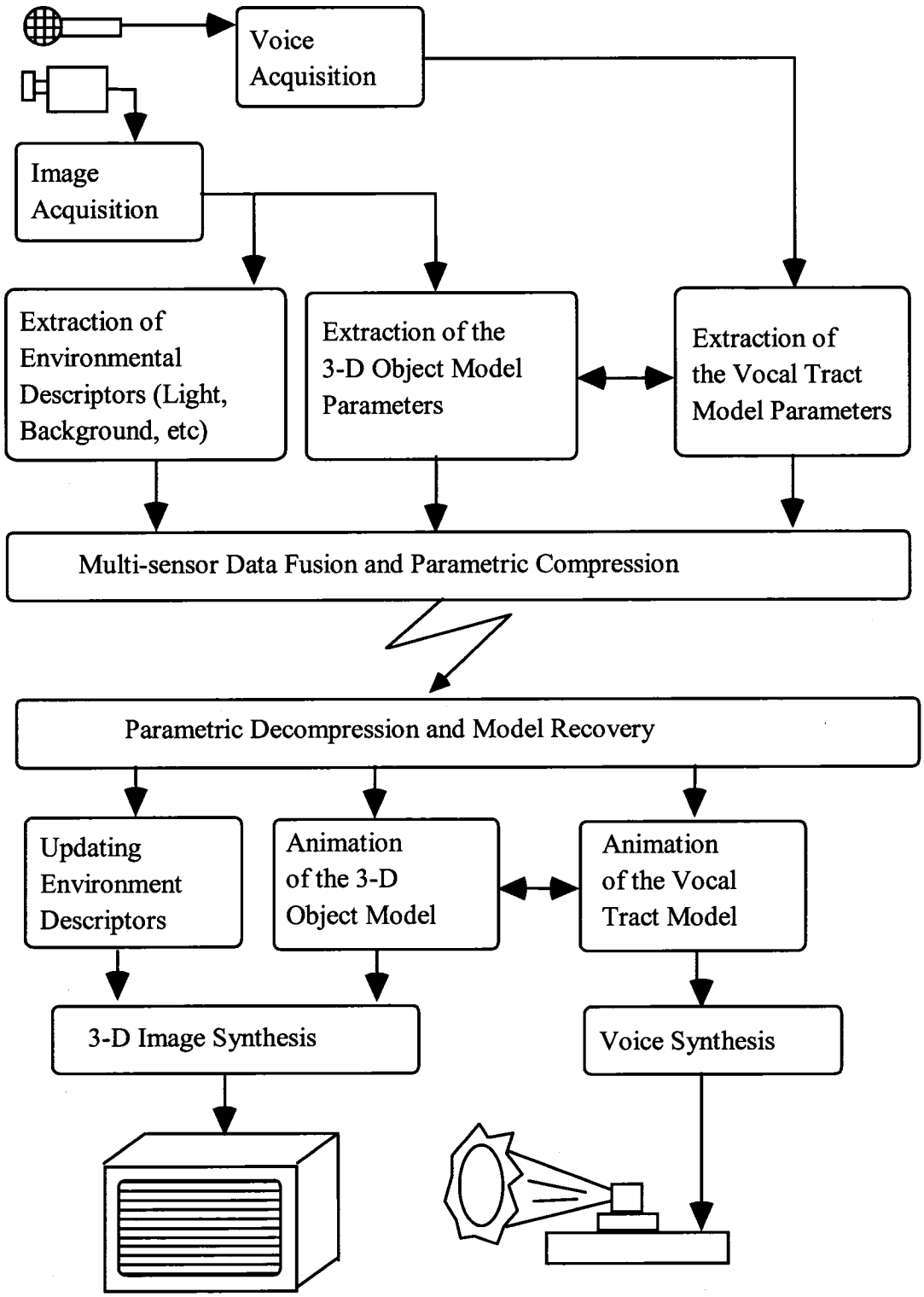


Fig. 1.1: Structural diagram of a one way videoconferencing system [Cor98].

Fig. 1.2 illustrates the simplified diagram of an MBVC showing examples of model extraction at the transmission end and animation using pose and expression parameters at the receiver end. Sending only change parameters greatly reduces the necessary bandwidth for visual communication. For instance, the data transfer rate required to transmit to the receiver the rigid and non-rigid motion of the face (3 translations, 3 orientations, and 4 expressions), at 10 frames/sec, is about 800 bps (8 bits/value × 10 values/frame × 10 frames/sec). As a comparison, for a sequence of images showing a moving object coded using a conventional compression scheme like MPEG-1, about 1,200,000 bps are required for reasonable picture quality at the same frame rate [Dev97].

While reducing the necessary bandwidth in videophone communication, the integration of this technology allows performing additional processing tasks such as face stabilization. Representing faces by 3D models allows focusing on head-and-shoulders, which permits using smaller display sizes. Another advantage of MBVC is the fact that the model used at the receiver (decoder) site may be different from the model used at the sender (encoder) site. "This feature allows avoiding a common objection to visual communication" [Dav97]. Another important advantage of MBVC is that the bit rate is independent of the image resolution and quality. The number of motion and model parameters to transmit over the communication channel does not change if the sender or receiver changes the image resolution. This is different from the classical interframe, block-based compression techniques such as MPEG1/2 and H.261/3 where the bit rate can increase significantly with the increase of the image resolution. Another issue of the current block-based compression methods is the motion of large objects. Such large objects are divided in blocks which are separately coded. This leads to an increase of the bit rate which is proportional to the moving area. In MBVC, the number of motion and model parameters does not depend on the size of the encoded objects. Another benefit of the MBVC over the classical compression techniques is object distortion in case of difficult tracking conditions. MBVC may place an object inaccurately in a frame due to a tracking failure but the object remains undistorted, which creates a better subjective feeling. These advantages make MBVC a serious candidate for applications in broadcast and super high definition video.

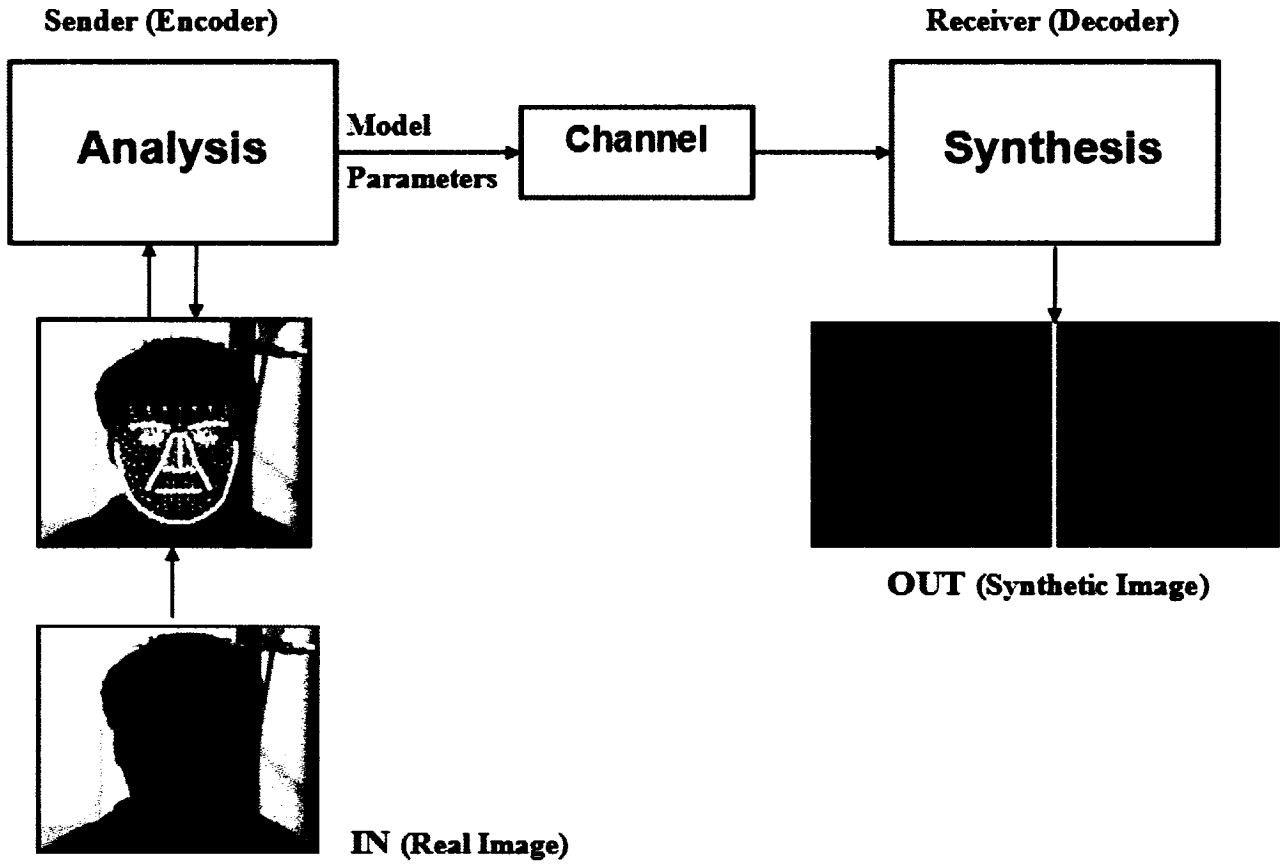


Fig. 1.2: Simplified diagram of a Model-Based Coder (facial image from [Pan05] database).

1.1 MODEL-BASED CODING ARCHITECTURE

Traditionally the MBVC research has concentrated in the areas of *Face Detection, Modeling, and Tracking (pose and expressions).*

1. *Face Detection* identifies and locates the face in first frames.
2. *Face Modeling* fits a 3D facial model to the input 2D facial image. The 3D model changes in shape and appearance to best fit the geometry and expressions of the input face. Facial texture is extracted, compressed and sent to the decoder.
3. *Face Tracking* extracts the 3D rigid motion of the head (or pose, or global motion), and computes the non-rigid facial motion (or expressions, or local motion) such as smiles and lip movements. Normally, the tracking should also continuously refine the face model by adding texture and changing the shape.

Face Detection

The first important step in a full automatic MBVC system is the identification and location of the face in first image frames. Because of variations in illumination, image resolution and representation (contrast, brightness, sharpness and color balance), background, orientation and facial expressions, the problem is very complex (Fig. 1.3.). Although most faces have similar structures with the facial features arranged in roughly the same spatial configuration, there is still a large variation due to non-rigidity and textural differences. There are significant geometrical or appearance differences even between images of the same person's face due to changes in expression and facial makeup.

Fig. 1.3: Challenges in Face Detection (from Lin04]).

Previous to this research, the author developed a method [Cor01], which uses a stochastic facial color model and the elliptical structure of the human head to detected faces. After separating the head from the background clutter, an ellipse is fitted to mark the boundary between the head region and the background. The work in this thesis employs for face and feature detection the method of Viola et al. [Vio01], which is currently the detection technique with the best speed/accuracy trade-off.

The next MBVC steps are face modeling and tracking covering global 3D-motion recovery, local motion estimation, expression and emotion analysis, and video-audio synchronization. The substantial effort

spent in these research areas is nevertheless justified since the recovery of the correct motion parameters is essential to the achievement of the ultra low bit rates promised by MBVC.

Face Modeling

Animating a face model in pose and expressions is a precondition of a MBVC functional system. Computer generated 3D models have extensively been used for the synthesis of naturally looking images and video. These models should have a limited number of vertices but enough details in order to distinguish facial features such as the lips or eyebrows. Rendering animated 3D models needs shape, texture and animation parameters data. A generic human facial shape is modeled as a 3D polygonal mesh. Acquired facial image texture is mapped onto the mesh to obtain a natural looking appearance. The shape and texture are usually considered together. The animation parameters describe the temporal variation of the model shape. For the specification of the facial expressions, we adopted the "muscle-based" facial model parameterization, which is not topology specific and can be controlled by a small number of parameters [Wat87]. This model incorporates three types of muscles: linear, sphincter, and sheet. Anatomical characteristics of the facial muscles and tissues are implemented as geometric distortion functions that describe the displacement of a predefined set of control points on the facial surface. Fig. 1.4 illustrates the steps of 3D face modeling: 3D model adaptation, texture mapping, and animation.

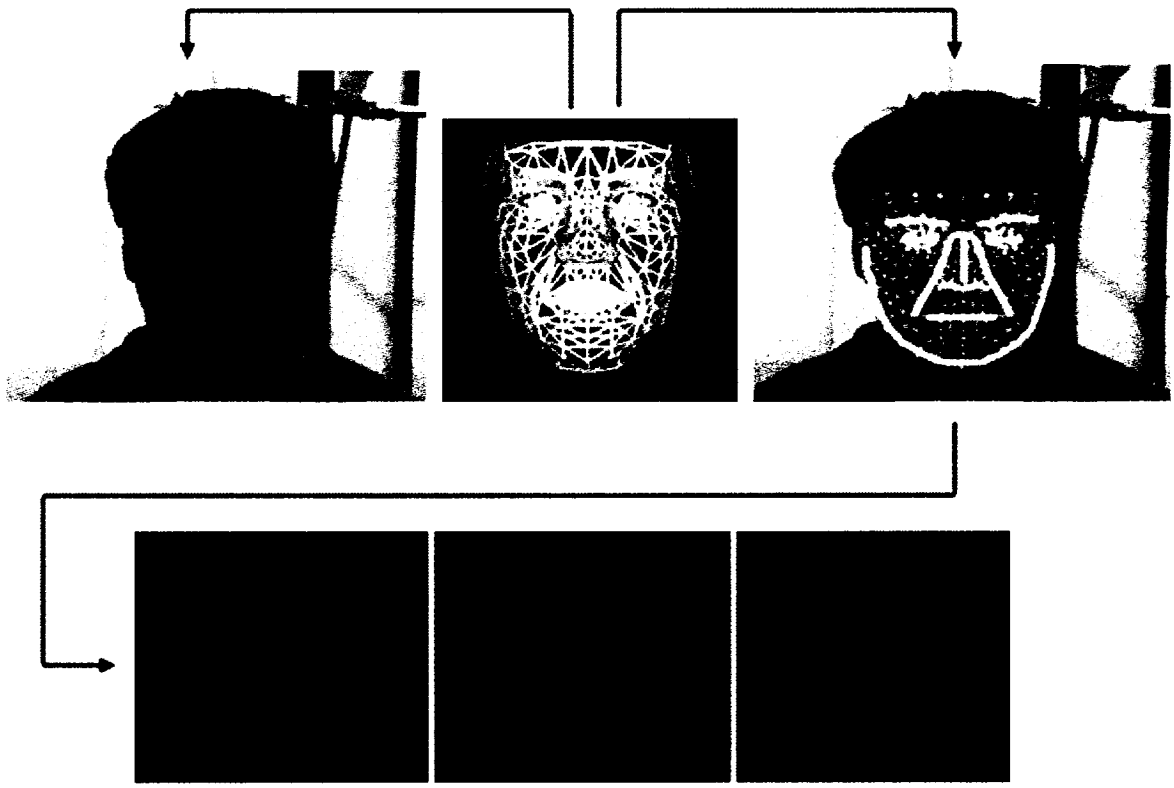


Fig. 1.4: Steps in Face Modeling (facial image from [Pan05] database).

The work on this thesis builds on the earlier research in facial animation of the author [Cor98] by enhancing the 3D muscle-based model with sets of constrained pair of muscles. The newly developed actuators deform a 3D generic model into specific 3D facial shapes and create only designated facial expressions. This thesis extends the Face Modeling area well beyond simple facial animation techniques, by explaining the facial deformation in a statistical framework. This allows modeling facial images in terms of controlled parameter sets, hence providing a natural and constrained basis for face segmentation and analysis.

Face Tracking

The face tracking problem is technologically difficult, as 3D motion parameters have to be extracted from a 2D video sequence. This can be done by tracking facial features or by using the optical flow in successive frames showing different points of view of the same 3D scene. In both cases the knowledge of the 3D scene and imaging process could be used to increase the accuracy and efficiency of the implementation. The effects of head motion and facial expressions are combined in these images, so it is

crucial to successfully separate the rigid from the non-rigid motion of the head (problem known as "pose/expression separation"). The head pose has to be accurately computed before attempting to recover the expressions.

In previous work the author proposed two algorithms for automatic head pose recovery using model-based approaches [Cor01], [Cor02]. Despite the 3D pose tracking method [Cor02] (Fig. 1.5.) was working at frame rate and showed good accuracy, it didn't include automatic techniques of initialization and recovery in case of failures. This thesis includes the initialization and recovery modules and augments the 3D tracking solution with facial expression recovery. The new tracking algorithm is formulated in a more robust framework, and its performance is validated in extensive experiments containing real and synthetic face image sequences, exposing large rigid motions and spontaneous expressions.

Fig. 1.5: Tracking the rigid motion of a human head (from [Cor02]).

1.2. CONTRIBUTIONS

This thesis represents a continuation of our work on model-based coding applications [Cor98], [Geo99], [Spo99], [Bon01], [Cor01], and [Cor02] addressing technologically difficult issues of face modeling and real time tracking for head pose and facial expression recovery in MBVC. In the earliest work [Cor98], the author developed a 3D wireframe head model and a deformation scheme capable of generating facial expressions. Subsequently, the research focused in areas of 2D Face Detection and Tracking [Cor01]. The model used to detect and track faces in 2½D (3D position and planar orientation) was a 2D

stochastic skin-color model constrained in shape by an elliptical, deformable head model. The motion estimation and filtering was achieved by a Linear Kalman Filter (LKF). Next, the author developed a 3D pose (3D position and 3D orientation) tracking method based on an Extended Kalman Filter (EKF), a 3D wireframe head model, and a correlation-based matching algorithm [Cor02]. In both tracking methods the input to the system was a 2D video sequence of the performer's head-and-shoulders. The output delivers the trajectories of tracked facial features, and the estimate of the 2½D or 3D motion of the head. Another element recovered by the 3D tracking method, besides the 3D relative motion object-camera, was the perspective camera focal length.

The ultimate goal of this thesis is to develop an Automatic Face Tracking System (AFTS) for a low bit-rate MBVC videoconferencing system. The specific thesis contributions are:

1. A new method for modeling the shape and appearance of human faces in 3D using a constrained 3D AAM. The method is an extension of the 2D AAM and uses a generic 3D wireframe model of the face, based on two sets of controls: the anatomically motivated muscle actuators to model facial expressions and statistically based anthropometrical controls to model different facial types (*3D Anthropometric Muscle-Based Active Appearance Model* (3D AMB AAM) (Chapters 2 and 3)
2. A new 3D tracking algorithm allowing real-time recovery of 3D position, orientation and facial expressions of a moving head. The method uses the newly developed 3D AMB AAM, a feature-based matching algorithm, and an Extended Kalman Filter (EKF) motion and expression estimator (Chapter 4 and 5)

An Automatic Face Tracking System was developed for applications that accept as input live video from camera or movie files. The tracking output (pose and expressions) is propagated to a 3D OpenGL window that animates the synthetic version of the performer's face (Chapter 5).

Our work in MBVC is similar in a few ways to the recent work of Ahlberg [Ahl98], [Ahl99], [Ahl03], and Strom [Str99], [Str02]. Ahlberg employs AAMs and the 3D Candide face model [Ryd97] for face tracking. The facial expressions are explained with the help of the MPEG-4 animation set, which is structure depended. Moreover, the expressions are handled separately from the texture formulation, and the system doesn't use a recursive motion estimator, which is crucial in attaining high tracking

performance. As a result the developed tracker works well only in constrained conditions, and often fails due to fast head motion. Work is under way to improve the speed and accuracy of the system.

Strom extends the work of Azarbayejani and Pentland in Structure-From-Motion (SFM) using a recursive EKF estimator [Aza95]. Strom enhances the SFM tracker robustness by including recovery in case of failure with the help of the same 3D Candide model. Additionally, the texture compression is formulated using Principal Component Analysis (PCA). However, Strom's system doesn't deal with facial expression recovery.

Our tracking system provides an original solution to the problem of SFM using an EKF [Cor02], which simplifies the computations during tracking. The tracking includes a novel 3D face modeling technique that statistically explains the variations of shape and appearance of human faces. The method uses a generic 3D wireframe model of the face, based on muscle actuators to model facial expressions and anthropometrical controls to model facial types. The areas of applications of the 3D AMB AAM extend beyond MBVC. Employing 3D AMB AAM in MBVC helps tracking with:

1. *Self-Occlusion Prediction*: the 3D model is used to predict feature occlusions, this way increasing the reliability of feature measurements.
2. *Tracker Initialization*: the features to be tracked are obtained by rendering the model in the estimated pose. This way rendered 3D patches help obtaining initial feature points and infer their occlusion during tracking.
3. *Texture Update and Compression*: a full 3D textured head model is recovered, and facial texture is PCA compressed.
4. *Facial Expression Recovery*: due to inherent AAM formulation the facial expressions are recovered at each frame during the synthesis step.
5. *3D Structure Constraint*: the 3D model fuses information of 3D structure and 2D measurements at each frame. The positions of the feature points affected by noise are constrained using the 3D model.

The developed EFK based tracking technique recovers 3D motion parameters, camera focal length, and facial expressions in a unified and recursive framework. The EKF provides a temporal estimation of model parameters including those related to facial expressions, this way providing robustness and stability over time.

1.3. THESIS OUTLINE

Chapter 2 of the thesis covers the head modeling section of the MBVC and presents a novel method for modeling the shape and appearance of human faces in 3D using a constrained 3D AAM. Chapter 3 shows that the developed face modeling technique method allows for accurate image fitting results. Extensive image segmentation experiments demonstrate the accuracy of the proposed algorithm against the classical 2D AAM. Chapter 4 describes a novel 3D tracking algorithm in a single-view video sequence. The method allows automatic and real-time recovery of 3D pose, and facial expressions of a moving head. The described method uses an Extended Kalman Filter to estimate the head pose (global motion) and the newly developed 3D AMB AAM to extract facial expressions (local motion). Chapter 5 describes extensive experiments to validate the tracking system on real and synthetic image sequences, and ground-truth data. The test sequences contain real and synthetic faces displaying large rigid motions and various expressions. Chapter 6 outlines the thesis contributions.

Chapter 2

Face Modeling Algorithm

Developing a face model is an essential step in developing a functional MBVC. Using a priori knowledge about the images to be transmitted, a 2D or 3D polygonal mesh model of a generic human face is fitted on the live image of a particular person as seen by the video camera at the emission end. Generally, the object fitting can be performed at low-level, using edge detection and region growing. These methods provide reasonable segmentations of fairly simple images but fail in the presence of structural complexity, noise, illumination variations and occlusions. Another segmentation approach, described in this thesis, models the class of objects of interest that is human faces, including a priori knowledge such as size, position, shape and appearance. This high-level information allows for a robust and accurate segmentation in uncontrolled environments.

This chapter describes a novel method for modeling the shape and appearance of human faces in 3D. Our work is an extension of the Active Appearance Models (AAMs), introduced by Edwards et al. [Edw98]. The AAM is a statistical model, based on the estimation of linear models of shape and texture variation. AAM was shown to be prolific in segmenting objects in low-resolution 2D images [Lan97], [Ste02], [Ste03], [Ers03], [Coo98], [Coo99], and [Tay98]. When modeling objects in high-resolution 2D and 3D images, this approach becomes infeasible due to excessive storage and computational requirements. Moreover, it was shown [Xia04], [Coo99], [Dav02], and [Dav03] that due to their representational capabilities, 2D AAM could generate illegal model instances. Also, classical AAM are not applicable to 6 degree-of-freedom (DOF) pose estimation. Our method incorporates the full 6 DOF pose and animation parameters as part of the minimization procedure. This allows for 3D pose estimation and facial expression recognition, without any restriction on face geometry. This chapter starts with a short history of models used for finding objects of interest in images, and then presents our solution to the problem of object segmentation.

2.1. DEFORMABLE MODELS

Because of its ability to fit objects of interest an active model is also a deformable model. Deformable models have been studied intensively in last years [Kas87], [Yul92], [Ter91], [Jai96], [Coo95]. They are similar to classical correlation models, except they have a built-in non-rigidity component. The flexibility component makes the deformable model active allowing for fitting the given data. A comprehensive but outdated survey of deformable models used in medical image analysis is found in [McI96]. Deformable models can be classified into two classes [Jai96]: free-form, and parametric models.

2.1.1. FREE-FORM MODELS

Free-form deformable models contain no knowledge about the object's shape except for some regularization constraints. Free-form models, such as "snakes" or Active Contours Models (ACM) match arbitrary shapes as long as regularization constraints (continuity and smoothness) are satisfied. Kass et al. [Kas87], [Ter88] were first to build an ACM to be used in image segmentation tasks. The snake's deformation is driven by a mix of three types of forces: the image, the internal and the external forces. The image force is minimized on matching image edges, the internal forces keep the shape smooth, and the external forces are manually controlled to initialize the snake to specific image features.

The original snake implementation uses only local information to find objects with no a priori knowledge of the searched object. This aspect makes it vulnerable to image noise and given initial position. Due to these flaws, the "snake" is suited only in interactive segmentation applications. Various attempts tried to improve the "snake" robustness [Coh91], [Cha94], [Ley93], [Ter91a]. In order to help the "snake" ignore isolated and weak image edges, and increase robustness to shrinkage, Cohen [Coh91] introduced a "balloon force" that can inflate or deflate the contour. Terzopoulos et al. [Ter91a] also included an inflation force that helps avoiding convergence on local minima.

2.1.2. PARAMETRIC MODELS

The parametric models include prior information on object's geometrical shape and deformation making them more constrained and robust than free-form models in image segmentation tasks.

Non-Statistical Models

Some deformable models are custom built for the task at hand. They use hand constructed parameterized curves and surfaces to model the non-rigid elements of objects. Yuille et al. [Yul92] use such a template to model face shapes and facial features (eyes, nose and lips). These parameterized curves and surfaces are fixed elastically to a global template frame to allow for minimal positional variations between facial features. The matching process attempts to align the template with pre-processed versions of the image, such as peak, valley and edge maps. An energy term controls the alignment by attracting the parameterized curves and surfaces to corresponding image features, while penalizing "deformation stress" in the template. This approach works for classes of objects with simple modes of deformation. In order to add complexity, some models may add a priori knowledge about the expected "physical" variation of the object. These models deform to match image evidence, such as edges, while constrained by physical properties [Ter91].

Statistical Models

The free-form model formulation is too general to achieve good image segmentation performance in uncontrolled environments. Also, non-statistical parametric models are too simple to represent complex structures such as human faces. These limitations led to segmentation techniques that utilize prior knowledge of object shape and deformation patterns. Some methods build on the existing free-form models, by adding statistical components. For instance, Szekely et al. [Sze96] developed elastically deformable Fourier parameterized models in 2D, called "Fourier snakes", and deformable Fourier surface models in 3D. They statistically derived the possible eigen-deformation of the training set to be used in medical image segmentation tasks. A well known class of statistical deformable models dealing with object boundaries is the Active Shape Model (ASM).

Active Shape Models

The Active Shape Model (ASM) was originally proposed by Cootes et al. [Coo95]. The technique uses a Point Distribution Model (PDM) to linearly model the shape, and an Active Contour Model [Kas87] deformation schema to interpret images. The advantage of ASM (also called "smart snakes") over the original "snakes" is that the model can only deform in ways found in a training set. ASM allow for significant variability but are at the same time constrained to the class of objects they represent. When used to interpret images the ASM search is initialized with a model guess such as the mean shape across the training database, to a position close to the target. The model deforms iteratively to minimize an energy function consistent with the prior knowledge about the object.

The first step in building an ASM is labeling the training database containing instances of the object of interest. Labeling denotes the action of annotating shapes, usually manually, with adequate landmark points. It is essential that the landmarks are placed in a way that preserves an exact correspondence between labels of training shapes. There are three types of landmarks that can be used in shape labeling [Coo95]:

1. Application-dependent or anatomical landmarks
2. Application-independent or mathematical landmarks
3. Pseudo or interpolated from the above types landmarks

Application-dependent landmarks in face images can be anatomically based points such as: centers of the eyes, nose tip etc. Application-independent landmarks are positioned based on geometrical or mathematical image information (i.e. points of high curvature, extremum etc.). Interpolated landmarks are points "in-between" type 1 and type 2 landmarks. They can be placed uniformly along a certain path, or using other strategies, and contribute the most to the shape boundary description (Fig. 2.1). Each shape is represented by a 2N element vector containing the x-y position of landmarks. Next step is to align the labeled training examples into a common coordinate frame by translating, rotating and scaling each training shape so that to minimize the sum of squared distances to the mean of the training set. The Principal Component Analysis (PCA) also known as Karhunen-Loeve (KL) transform is used to find the main axes of variation

in the aligned training set. This way only the first few modes of variation are kept, which account for the majority of the variation. The derived shape model is:

$$x = \bar{x} + Up_x \tag{2.1}$$

Where $\bar{x}$ is the mean of the aligned training examples, U is a $2NxP$ matrix whose columns are unit vectors along the principal axes of the shape space, and p_x is a $P < N$ element vector of shape parameters. New shape examples are obtained by varying the shape parameters within limits learnt from the training set. The trained shape model is a knowledge-based method to find object boundaries using priors about the searched shape. Basically, ASM is a deformable template that imposes stronger priors than snakes. Assuming a correct initial estimate for the pose and shape parameters, the ASM iterative fitting algorithm is as it follows:

1. Start with an initial shape (usually the mean shape) placed near the object of interest
2. Find the best local appearance match at each landmark along its normal
3. Update the pose and shape parameters so that the model instance best fits the found points
4. Repeat until convergence

The search performance can be improved in speed and accuracy by using a multi-resolution ASM. That is the search starts on a coarse level of a gaussian image pyramid, and is gradually refined.

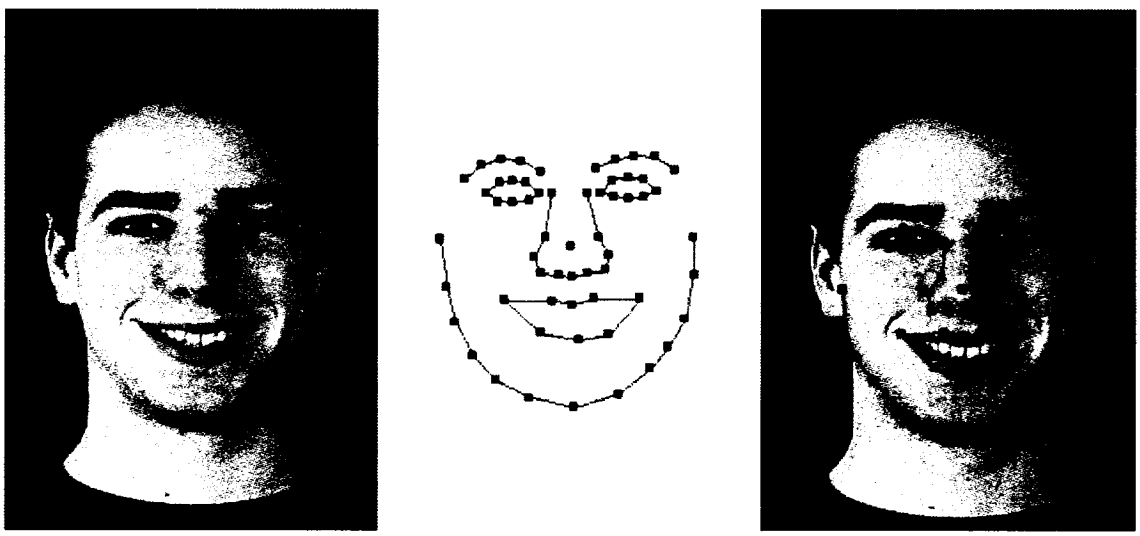


Fig. 2.1: Labeling facial shapes using a point model (facial image from [Pan05] database).

Shape or pose variability among objects of the same class, non-uniformity of landmark registration across the training set or simply human induced errors during the labeling process confer a non-linear behavior to the shape model. This aspect leads to creation of "illegal" shape instances that is shapes not seen in the training set.

Non-linear ASM extensions have been proposed. Edwards et al. [Edw98a] built a non-linear ASM built with the help of a Multi-Layer Perceptron. Romdhani et al. [Rom99] proposed a non-linear shape model to address the problem of large facial pose variation, by employing Kernel Principal Component Analysis (KPCA) [Sch97].

Active Appearance Models

The ASM does not use the texture information when locating the shape of the modeled object class. Related to ASM, the AAM were introduced [Edw98]. The AAM fitting method can match a full shape and grey-level appearance to a target image. The AAM were proved to be successful in medical image analysis [Coo98], [Coo01], facial recognition [Coo00], [Edw98], [Kan02], high quality talking heads applications [Dev01], [The01], [Hac03], face tracking [Edw98a], [Ste01], [Ahl03], and age-progression simulation [Lan99].

2.2. 2D ACTIVE APPEARANCE MODELS (2D AAM)

Generally, AAM refers to a specific shape-appearance model and an algorithm for fitting the model to images showing objects of the modeled class.

2.2.1. MODEL PARAMETERIZATION

AAM is similar to Eigenfaces [Tur91], which is a representation resultant from Principal Component Analysis (PCA) applied only to the facial texture. Eigenfaces describe the principal modes of variations of facial images. An image containing N pixels may be viewed as a vector in N dimensional space. The collection of faces represented as vectors forms the original space of facial images. However, all face vectors are clustered in a low-dimensional subspace, named *face*

space. The original face space is transformed into a low-dimensional space using PCA. The PCA projection forms an orthonormal basis for the face space, and it is basically a *lossy* dimensional reduction method. This low-dimensional eigenspace is estimated by finding the eigenvectors of the covariance matrix built from training images. The eigenface space is built in few steps:

1. Collect P facial images and align them using two-point warping (e.g. all face images have the eyes in the same locations). Also a fixed facial region is defined to help collecting pixels in training vectors. This way each face can be stored in a vector of size N:

   $$g^i = [g_1^i \quad g_2^i \quad \cdots \quad g_N^i]^T . \tag{2.2}$$

   This step is also called *normalization.* Fig 2.2 illustrates example of images "eye-to-eye" geometrically normalized.

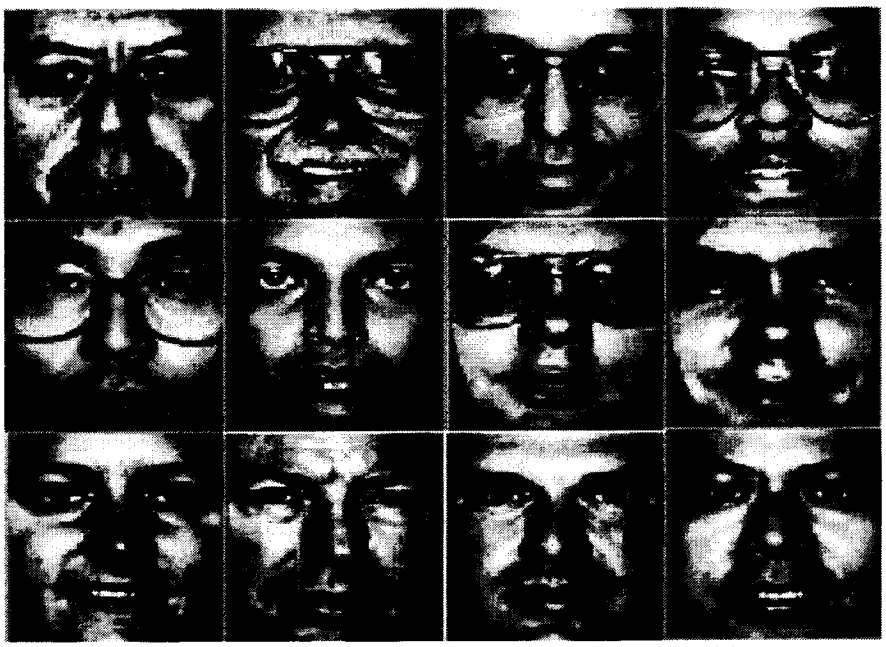


Fig 2.2: Aligned facial images (from FERET dataset [Phi99])

2. Mean center the training data:

$$\bar{g}^i = g^i - \frac{1}{P}\sum_{i=1}^{P} g^i \tag{2.3}$$

3. Concatenate the normalized face vectors in a matrix of size $N \times P$:

$$\bar{G} = \left[\bar{g}^1 \,|\, \bar{g}^2 \,|\, \ldots \,|\, \bar{g}^P\right] \tag{2.4}$$

4. Perform an eigen decomposition of the covariance matrix:

$$C = \bar{G}\bar{G}^T = \Omega\Lambda\Omega^T \tag{2.5}$$

where matrix Ω contains M eigenvectors ($M \leq P \ll N$), that is the basis of the face space. The eigenvectors are sorted high to low corresponding to the eigenvalues stored in matrix Λ. The set of unit length eigenvectors compose an orthonormal basis where the first direction represent the direction of maximum variance in the images, the second direction represent the next highest variance, and so on.

5. If we collect the eigenvectors in a matrix U, any face g can be parameterized in the M dimensional space as a vector p_g:

$$p_g = U^T(g - \bar{g}) \tag{2.6}$$

Also, faces can be synthesized or reconstructed from the compressed face space to the original face space:

$$g = \bar{g} + Up_g \tag{2.7}$$

The dimension of the eigenface space is less or equal to the number of training images $M \leq P$. Normally the size of the training set is much smaller than the size of each image $P \ll N$. From (2.6) and (2.7) we obtain:

$$\hat{g} = \bar{g} + UU^T(g - \bar{g}) \tag{2.8}$$

and the reconstruction error:

$$e(g) = \|\hat{g} - \bar{g}\| \tag{2.9}$$

The closer a face is to the face space the smaller the reconstruction error is (Fig. 2.3a). This means that a face image not present in the training set can be reconstructed with a low error only if it is similar to the training images.

Fig 2.3: Reconstructed facial images (correct and erroneous instances).

There is a problem with the above formulated eigenspace. Due to the simple two-point alignment method the resulted eigenspace will be over-complete. That is, the space can synthesize faces that do not belong to the face space (e.g. face with multiple noses, chins etc.). Fig 2.3b shows a face whose height was larger that the one present in the training set. As a result, the face is reconstructed without the mouth. One solution to deal with this aspect is to use non-linear methods to model the face set. Another approach is to use other alignment techniques that will generate better normalized faces. Such technique is Generalized Procrustes Analysis (GPA) [Gow75], which aligns in a least-squares sense the facial contours into a normalized face shape.

AAM supersedes the Eigenface method by combining the facial shape and pixel intensity information into one formulation. AAM learns from a training set to create a compact parameterization of the object's properties variability. Typically, the modeled properties are shape and texture. Object shape is defined by manually or automatically labeling each image example with landmarks. The shape examples are then aligned to a common mean using GPA. This geometrical normalized frame represents the *shape-free* reference where the texture samples are warped to and sampled from. After geometrical normalization, PCA is used to model shape and texture variability. Employing prior knowledge of the optimization space, the obtained models can be fitted to unseen images, given a close-to-target search initialization.

2.2.2. MODEL FITTING

The fitting process uses the appearance model described above and an Analysis-by-Synthesis (ABS) technique to interpret images, given a reasonable starting position. The fitting algorithm generates synthetic examples to match the given images by efficiently adjusting the model parameters. Fitting an AAM to an image is a non-linear optimization problem. The basic

approach starts with generating different model instances by using incremental updates to the shape, appearance and pose parameters. The process continues by computing an error image between the model instance and the input image. The basic AAM fitting technique assumes a constant linear relationship between the error image and the incremental updates to the parameters.

Baker and Matthews [Bak01] indicate that the basic additive method of fitting scheme is incorrectly used due to the non-linear nature of the process. They propose an alternative inverse compositional image alignment (ICA) algorithm, which claimed to be slightly faster and more precise than the basic one. The proposed method can be applied only when shape and appearance are parameterized separately.

Hou et al. [Hou01] proposed Direct AAM (DAAM), which defines a fitting technique that predicts shape directly from texture when the two are sufficiently correlated.

Cootes et al. [Coo98a] also proposed an alternative fitting algorithm in which the residuals correct pose and shape instead of appearance parameters. Cootes et.al. [Coo02] compared the basic searching method with the three variants described above [Bak01], [Hou01], [Coo98a]. They found that the simple linear AAM update scheme with a minor modification outperformed the three variants. The modification of the search algorithm consists in performing at least one "forced" iteration regardless of the minimization result.

2.2.3. CHALLENGES

Some of the challenges of 2D AAM applied to object segmentation and recognition are: viewpoint changes, accuracy and speed. The major drawback with ASM and AAM models is that they employ PCA, a linear technique, to model deformations that is a non-linear process. The non-linearity is introduced by object curvature, pose, or during the alignment and construction of the PDM. The linear modeling of 2D AAM assumes that the training set will generate a uniform cluster in shape-appearance space. However, as more samples are added to the training set, the built model will generate multiple, separate clusters in the shape-appearance space. The result is

an unreliable model that affects the performance in segmentation, tracking and recognition applications. There are two major sources that break the linear assumption of 2D AAM training set [Dav03]:

1. Variations in face orientation
2. Incorrect shape labeling

There are three approaches that deal with the rotational non-linearity [Coo00]:

1. the use of a 3D model [Bla99], [Sko03], [Mit02], [Bux01], [Rom03]
2. integrating the non-linearity into a 2D model [Rom99], [Mur95]
3. multi-view 2D AAM [Coo00]

One solution to the pose variation problem is the use of the 3D AAM. Usually, in order to build a 3D AAM, researchers use data sets of faces acquired with a 3D scanner [Bla99], [Sko03], [Bux01]. Blanz et al. [Bla99] describe the 3D Morphable Model (3DMM) of human faces, built from a high resolution 3D polygonal wireframe mesh (76000 vertices). Their model is similar to the AAM, but uses separate models for shape and texture. Other researchers [Sko03], [Bux01] built a dense correspondence over a set of training shapes of the human face starting with a set of few manually annotated landmarks. Also, Mitchell et al. [Mit02] built a 3D AAM from volumetric cardiac magnetic resonance images, and used it for segmentation and recognition applications. The process of building 3D AAMs, from collecting data to image segmentation requires extensive work, complex algorithms and high computational power. For instance the fitting process takes 30 seconds per frame in [Rom03].

Research has been conducted to correct the problem generated by incorrect shape labeling. In 2D, correspondence is set using manually labeled landmarks, but this is a subjective, time-consuming, and error-prone process. By carefully selecting landmark points by hand, a near optimum labeling can be achieved which will minimize the non-linearity of a training set.

Heap and Hog define a Cartesian-Polar Hybrid PDM [Hea95] that uses polar coordinates to model deformations more accurately. Sozou et al [Sou95] employ a multi-layer perceptron to perform non-linear principal component analysis on the training set. The network architecture is application specific, and training is a time consuming process.

Davis et al. [Dav02], [Dav03] show that the correspondences found using manual labeling methods are erroneous and generate inadequate models. They define the "best" shape model the one with optimal compactness, specificity and generalization properties. Their method finds a pattern of "legal" variation in the shapes for a given class of images, providing this way a compact representation of shape. The optimum model is built automatically using an objective function based on the minimum description length principle (MDL) [Ris83].

The model parameterization should express a trade-off between general and individual. Generally, when an observed data class expose large generality, it is likely to generate invalid members of the class. Recently 3D model-based approaches to the image segmentation containing deformable objects have been described. One motivation of using 3D model-based approaches to the image interpretation is achieving higher performance by constraining the solutions of the modeled object to valid instances of the 3D model. Xiao et al. [Xia04] proposed a combined "2D+3D" AAM to fit 3D shapes to images. They showed also that due to their representational power, 2D AAM could generate illegal model instances that are not physically attainable. Their method describes how 3D modes can be used to constrain the AAM so that it can only create valid model instances.

2.3. A 3D ANTHROPOMETRIC MUSCLE-BASED AAM (3D AMB AAM)

This chapter describes a method that tries to overcome problems encountered by classical AAM. We use a generic 3D wireframe face model to solve the storage, pose and labeling related problems. Classical AAM parameterizes the facial shape in 2D or 3D space defined by the labeled landmark values. We take a different approach, in order to have total control of generated model instances and increase recognition accuracy, stability and speed.

The shape of our geometric model is linearly controlled by articulator parameters that control the facial expressions and facial types. Continuous changes in the model shape parameters generate smooth transitions between expressions and facial types, thus constraining our geometrical search space. We choose two modeling techniques to control the 3D facial shape deformation:

1. Muscle-Based modeling to represent facial expressions
2. Anthropometry-Based modeling to represent different physiognomies

Muscle-Based uses a muscle animation model to generate facial expressions. Anthropometry-based method uses a constrained underlying muscle animation model. The modeled shape parameters (expressions and physiognomies) are application depended. For example, a lip-reading application employs deformations describing lip motion, and discards those tracking the eyebrows. The model appearance is considered the texture map of the 3D model vertices. The developed model-based approach also provides the base for a wide range of applications. For instance, applied to facial analysis model-based approaches can explain the face pose, expression or identity. The next sections introduce the employed 3D modeling techniques and the model training process used to derive model parameterization and fitting algorithm.

2.3.1. MUSCLE-BASED FACE MODELING

The facial shape is determined mainly by the frontal part of the skull and facial muscles. The facial muscles are attached to the bones of the skull in the subcutaneous tissue. The muscles responsible for generating facial expression are located around the mouth, eyes and nose, acting as dilators and sphincters (Fig.2.4).

For facial animation we have adopted Waters' muscle-based face model [Wat87], [Par94] together with the Facial Action Coding System (FACS) [Ekm86]. This model provides a simplified version of the physically based animation using only facial muscles for the activation of the neighboring nodes in a facial mesh. The muscles have vector properties and are independent of the underlying bone structure. This results in a model independent of the facial topology. Other advantages of the muscle-based model are: a reduced set of facial parameters (muscle activation values) convenient for low-bit rate communication, and low computation time to generate facial expressions. A well-known muscle-based coding system, largely used in the facial modeling field is the FACS. Facial animation field describes face expressions based either on MPEG-4 SNHC-standard [Tek00] or FACS. The MPEG-4 coding standard is useful for animating facial avatars, but lacks in modeling facial behavior. For instance, MPEG-4 does not encode behaviorally characteristic facial deformations and texture changes (wrinkles, bulges etc).

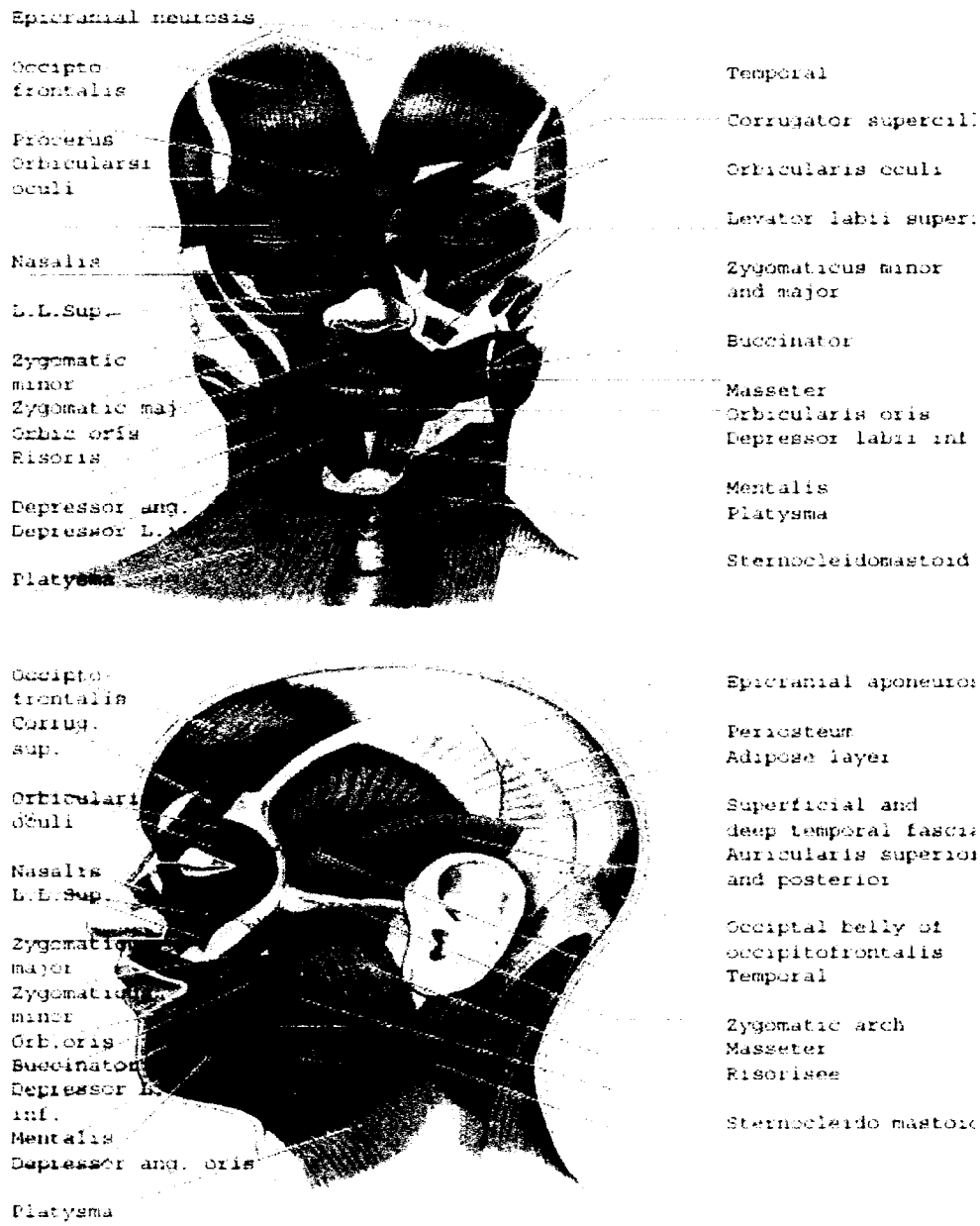


Fig. 2.4: Facial muscles [Moo99].

Facial Action Coding System

Interest in facial expression can be dated back to the mid 19th century, when the most influential theorist Charles Darwin wrote *The Expression of the Emotions in Man and Animals* [Dar896]. Darwin believed that facial expressions have universal and native characteristics. Later, two sign communication psychologists, Ekman and Friesen, developed the anatomically oriented FACS in 1977 [Ekm77], and updated it later [Ekm86]. Ekman and Friesen derived FACS based on

numerous experiments with facial muscles. The experiments used electrical simulation on individual muscles and observed the correlation with changes on facial appearance. They defined the Action Unit (AU) as a basic visual facial movement, which cannot be decomposed into smaller units. One or more muscles control the AUs. Their system also describes complex facial expressions in terms of basic facial muscle actions. Ekman and Friesen reduced the distinguishable expression space to an ample system, which could distinguish all possible visually facial expressions by using only 44 AUs. The six basic emotional or prototypic expressions [Ekm71]: happiness, surprise, sadness, fear, anger, and disgust (Fig. 2.5), and complex facial expressions can be obtained by combining different AUs. Recently, Ekman, Rosenberg, and Hager built Facial Action Coding System Affect Interpretation Dictionary (FACSAID) stored in a relational database, which relates facial expressions to their psychological interpretations.

Fig. 2.5: The six prototypical facial expressions: disgust, fear, happiness, surprise, sadness, and anger, from [Kan00].

The updated version of FACS summarized into the *Manual for the Facial Action Coding System* "was written in a self-instructional format, to serve as an initial tutor and subsequently as a reference in scoring facial behavior" [Ekm86]. The manual specifies 33 AUs in terms of its muscular basis, appearance, and instructions to perform the movement (Table 2.1). Additionally, there are 11 simpler AUs, which don't involve muscle actions (Table 2.2).

AU	FACS Name	Muscular Basis
1	Inner Brow Raiser	Frontalis, Pars Medialis
2	Outer Brow Raiser	Frontalis, Pars Lateralis
4	Brow Lowerer	Depressor Glabellae; Depressor Supercilli; Corrugator
5	Upper Lid Raiser	Levator Palpebrae Superioris
6	Cheek Raiser	Orbicularis Oculi, Pars Orbitalis
7	Lid Tightener	Orbicularis Oculi, Pars Palebralis
8	Lips Toward Each Other	Orbicularis Oris
9	Nose Wrinkler	Levator Labii Superioris, Alaeque Nasi
10	Upper Lip Raiser	Levator Labii Superioris, Caput Infraorbitalis
11	Nasolabial Furrow Deepener	Zygomatic Minor
12	Lip Corner Puller	Zygomatic Major
13	Cheek Puffer	Caninus
14	Dimpler	Buccinnator
15	Lip Corner Depressor	Triangularis
16	Lower Lip Depressor	Depressor Labii
17	Chin Raiser	Mentalis
18	Lip Puckerer	Incisivii Labii Superioris; Incisivii Labii Inferioris
20	Lip Stretcher	Risorius
22	Lip Funneler	Orbicularis Oris
23	Lip Tightner	Orbicularis Oris
24	Lip Pressor	Orbicularis Oris
25	Lips Part	Depressor Labii, or Relaxation of Mentalis or Orbicularis Oris
26	Jaw Drop	Masetter; Temporal and Internal Pterygoid Relaxed
27	Mouth Stretch	Pterygoids; Digastric
28	Lip Suck	Orbicularis Oris
38	Nostril Dilator	Nasalis, Pars Alaris
39	Nostril Compressor	Nasalis, Pars Transversa and Depressor Septi Nasi
41	Lid Droop	Relaxation of Levator Palpebrae Superioris
42	Slit	Orbicularis Oculi
43	Eyes Closed	Relaxation of Levator Palpebrae Superioris
44	Squint	Orbicularis Oculi, Pars Palpebralis
45	Blink	Relaxation of Levator Palpebrae and Contraction of Orbicularis Oculi, Pars Palpebralis
46	Wink	Orbicularis Oculi

Table 2.1: Anatomical basis of each AU (AU2 and AU40 are not defined).

AU	FACS Name
19	Tongue Out
21	Neck Tightener
29	Jaw Thrust
30	Jaw Sideways
31	Jaw Clencher
32	Lip Bite
33	Cheek Blow
34	Cheek Puff
35	Cheek Suck
36	Tongue Bulge
37	Lip Wipe

Table 2.2: Miscellaneous AUs.

Since FACS itself is purely descriptive, emotion specified expressions are not part of it. Hence, Ekman and Friesen also developed *Emotion FACS* (EMFACS) [Ekm83], which is a method of combining FACS scores into actions that encode emotion specified expressions (e.g. happiness or surprise). For instance a smiling appearance, common in "happiness" emotion, in which the lip corners are pulled upward and laterally, can be provoked by the combined action of Zygomatic Major (AU12), and Orbicularis Oculi (AU6) muscles.

Our 3D model employs the muscle-based technique to model both the expression and anthropometric spaces. In order to differentiate between the two spaces, we define Expression Action Unit (EAU) to represent the facial expression space, and Anthropometry Action Unit (AAU) to represent the anthropometric space.

Muscle-Based Model

In Waters' facial animation scheme, muscles can be described as vectors with direction and magnitude in two and three dimensions. The direction is headed to the attachment point on the bone, and the displacement of the deformation depends on the muscle elastic constant and the tension of the muscle contraction. The model mimics at a simple level the action of three primary muscle groups of the face (Fig. 2.6):

1. *Linear muscle*, such as the *Zygomatic Major*
2. *Sphincter muscle*, such as the *Obicularis Oculi*
3. *Sheet muscle*, such as the *Frontalis Major.*

It was shown [Wat87] that a only 8 pairs of facial muscles (16 muscles) plus the jaw rotation, eyelids, and eyes motion parameters, mapped to AU contractions, can practically cover an extensive space of human facial expressions. The 3D model used in this research implements 7 pairs (left/right) of muscles and the jaw rotation (Fig. 2.6):

1. Zygomatic Major
2. Anguli Depressor
3. Frontalis Inner
4. Frontalis Outer
5. Frontalis Major
6. Lateral Corrugator
7. Nasi Labii
8. Jaw Drop

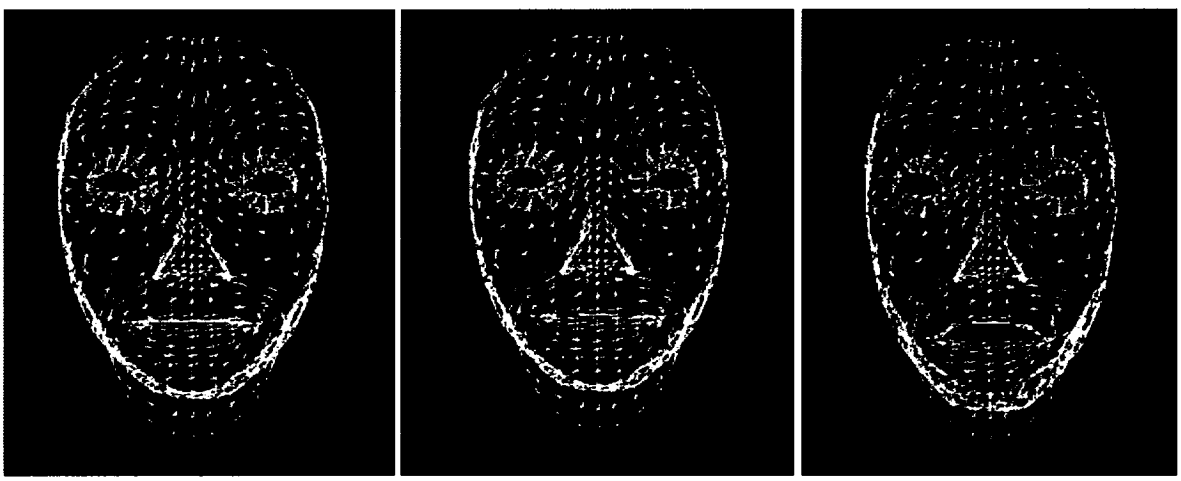


Fig. 2.6: The 3D Muscle-Based Face Model.

Each muscle activity is mapped to an AU. In our research we use six AUs to derive facial animation modes (Fig 2.7):

1. **Jaw Drop** or **AU 26**, present in emotional expressions such as: surprise, happiness, fear, and disgust.
2. **Lip Corner Puller** or **AU12**, caused by *Zygomatic Major* muscle. AU12 is a major component of happiness and surprise expressions.
3. **Lip Corner Depressor** or **AU15**, caused by *Anguli Depressor* muscle. AU15 is present in sadness and disgust expressions.
4. **Inner Brow Raiser** or **AU1**, caused by *Inner Frontalis* muscle. AU1 is part of sadness, surprise and fear expressions.
5. **Outer Brow Raiser** or **AU2**, caused by *Outer Frontalis* muscle. AU2 is present in surprise, fear and disgust expressions.
6. **Brow Lowerer** or **AU4**, caused by *Corrugator* muscle. AU4 is mainly part of anger and disgust expressions.

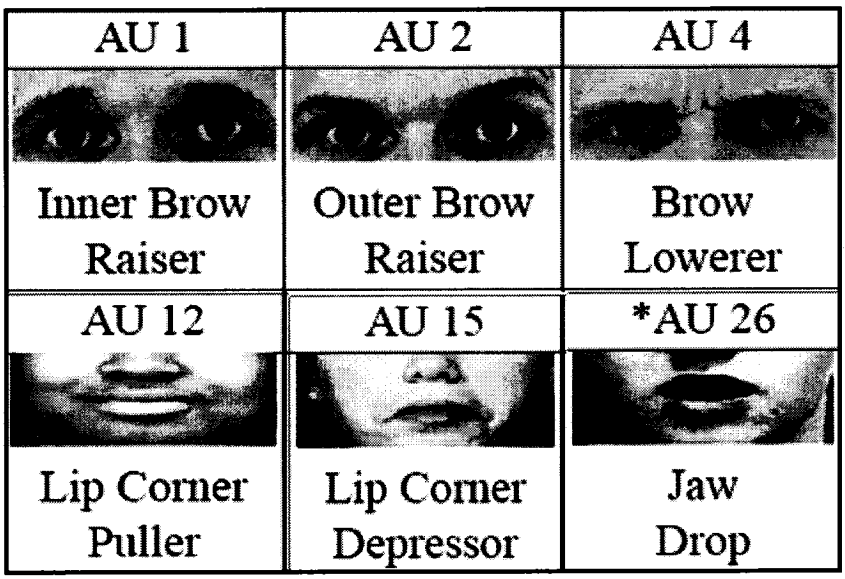


Fig. 2.7: The modeled AUs [Ekm86]

AUs occur alone or in combinations. The combinations may be additive, in which case the appearance caused by each AU is preserved, or non-additive, in which case the appearance caused by each AU is changed. An example of an additive combination is smiling (AU12) and jaw drop (AU26) which often occurs in "happiness" expression. Non-additive interactions also called co-articulation effects are difficult to recognize and their exposed appearance is highly

dependent on timing. An example of non-additive combination is AU12+15, which is present in "embarrassment" expression [Tia03].

In order to better control the muscular activity during the fitting process and eliminate non-additive combinations, we built context-depended AUs, and called them Expression Action Units (EAUs). First of all, the new muscles are composed of symmetrical left/right facial muscles, which deform simultaneously. Secondly, they represent combinations of the implemented AUs or pairs of muscles that pull in opposite directions. For example Lip Puller and Lip Depressor can cancel each other while making an expression. In order to avoid the co-articulation effects, a combination of the two muscles, named Mouth Corners, can activate one AU at the time. The new muscle behaves as AU12 when pulling upwards, for values between [0, 1] or AU15 when pulling downwards, for values of [-1, 0]. This way the newly developed EAUs allows only additive combinations:

1. **Jaw Drop (AU26)** remains unmodified
2. **Mouth Corners (AU12, AU15)** is composed of Left /Right Zygomatic Major and Left /Right Anguli Depressor
3. **Eyebrow Middle (AU2, AU4)** is composed of Left /Right Frontalis Major and Left /Right Corrugator
4. **Eyebrow Inner (AU1, AU4)** is composed of Left /Right Frontalis Inner and Left /Right Corrugator
5. **Nose Base (AU9)** is composed of Left /Right Nasi Labii

Using additive combinations of muscle actuators we also implemented the six basic emotional expressions. The combinations containing the muscles responsible for generating these expressions, and their range of deformation are shown in Table 2.3.

Emotional Expressions			
1.	Happiness 	Mouth Corners	0.0 – 0.9
		Eyebrow Inner	0.0 – 0.2
		Jaw Drop	0.0 – 0.5
2.	Anger 	Nose Base	0.0 – 0.7
		Eyebrow Inner	-0.9 – 0.0
		Eyebrow Middle	-0.9 – 0.0
		Eyebrow Outer	-0.5 – 0.0
3.	Surprise 	Mouth Corners	0.0 – 0.4
		Eyebrow Inner	0.0 – 0.5
		Eyebrow Middle	0.0 – 0.6
		Jaw Drop	0.0 – 0.9
4.	Sadness 	Mouth Corners	-0.9 – 0.0
		Eyebrow Inner	0.0 – 0.7
		Eyebrow Middle	0.0 – 0.5
5.	Fear 	Eyebrow Inner	0.0 – 0.2
		Eyebrow Middle	0.0 – 0.5
		Eyebrow Outer	0.0 – 0.2
		Jaw Drop	0.0 – 0.4
6.	Disgust 	Nose Base	0.0 – 0.9
		Eyebrow Inner	- 0.1-0.0
		Eyebrow Outer	0.0 – 0.2
		Jaw Drop	0.0 – 0.1

Table 2.3: The implemented EMFACS model.

The implemented emotional expressions are illustrated at their peak in Fig. 2.8.

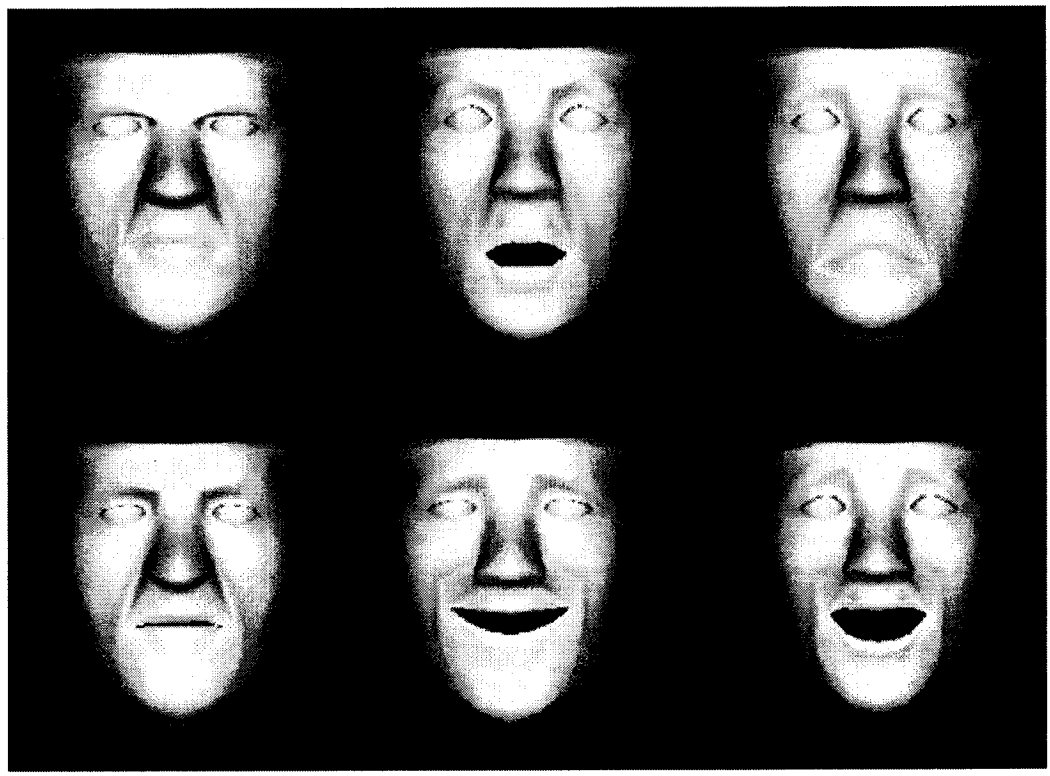


Fig. 2.8: The six implemented emotional expressions at their peak

2.3.2. ANTHROPOMETRY-BASED FACE MODELING

Anthropometry, the measurement of living subjects, starts with the identification of anatomical landmarks on a subject's face or skull. Anthropologists and plastic surgeons use these landmarks to describe the human face. Forensic anthropology uses anthropometry data to reconstruct individuals' appearance from their remains. Anthropometric measurements are also used in plastic and reconstructive surgery planning and assessment [Far94]. It was shown [Far94], [Kol96] that repetitive measurements of the same individual are consistent, and can be successfully used to differentiate among individuals [DeC98].

Farkas [Far94] and Kolar [Kol96] define a large set of craniofacial landmarks and measurements to describe the human face (Appendix A). A summary of these are illustrated in Fig. 2.9. The landmarks are represented by abbreviations of corresponding anatomical terms. Two landmarks pairs *(t, or)* in either side of the head define a horizontal plane known as Frankfurt horizontal (FH) orientation of the head [Far94], [Kol96]. The vertical axis is defined by the landmarks *n*

(*nasion,* skull feature between the eyebrows) and *sn* (*subnasale*, the center of the nose base). The craniofacial measurements are made with respect to this 3D local coordinate system. A detailed description of the anthropometric landmarks and measurements is presented in Appendix A. [Kol96]. Farkas [Far94] defines five types of facial measurements (Fig. 2.9 b):

1. the *shortest distance* between two landmarks (i.e. *en-ex*, the distance between eye corners)
2. the *axial distance* between two landmarks, measured along one of the axes of the coordinate system, with the head in FH position (i.e. *v-tr*, the distance between top of the head and hairline)
3. the *tangential distance* between two landmarks, measured on the surface of the face along a path (i.e. *ch-t*, the surface distance from the corner of the mouth to tragus)
4. the *angle of inclination* between two landmarks with respect to one of axes (i.e. angle between ear axis and vertical axis)
5. the *angle between locations* (i.e. the angle at the chin)

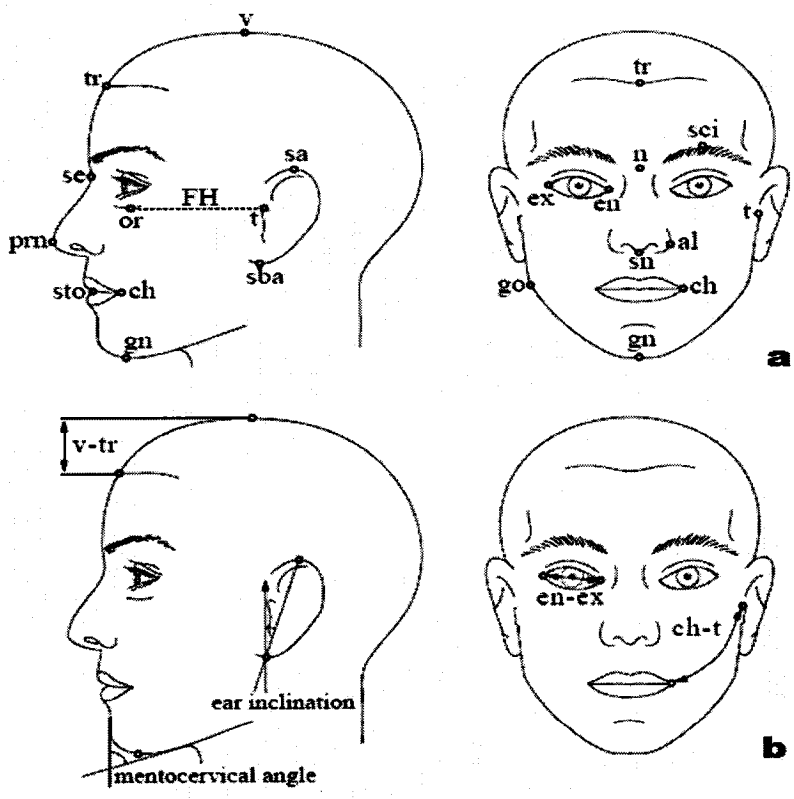


Fig. 2.9: Anthropometric facial landmarks (a) and measurements (b) [Far94].

This thesis employs anthropometry to build and deform the 3D face model. The AAUs, chosen in conformity with anthropometrical statistics, provide a way to deform the 3D generic mesh into a personal one. The AAUs are deformation parameters used to make the jaw wider, the nose-bridge narrower, etc. We annotated our 3D model using predefined anatomical landmarks, which are used to generate 11 facial measurements (see Appendix A.). We name them Anthropometric Action Units (AAUs):

1. Cranial width (*t-t*)
2. Facial height (*tr-gn*)
3. Facial depth (*or-t*)
4. Mandible width (*go-go*)
5. Chin height (*sl-gn*)
6. Nose height (*n-sn*)
7. Nose bridge length (*n-prn*)
8. Nose width (*al-al*)
9. Mouth vertical shift (*sn-sto*)
10. Mouth width (*ch-ch*)
11. Eyebrow height (*sci-ex*)

In order to keep the AAUs independent of the facial mesh structure, we built them from symmetric pairs of constrained facial muscles. Similarly to the animation scheme, the anthropometric actuators are anchored in the anatomical landmarks. We constrained the deformation to allow the displacement in one dimension only. For example the "Nose width" locally deforms the nose shape only on horizontal direction. Fig. 2.10 illustrates few instances of the 3D-AMB model obtained by varying anthropometric and muscle parameter values.

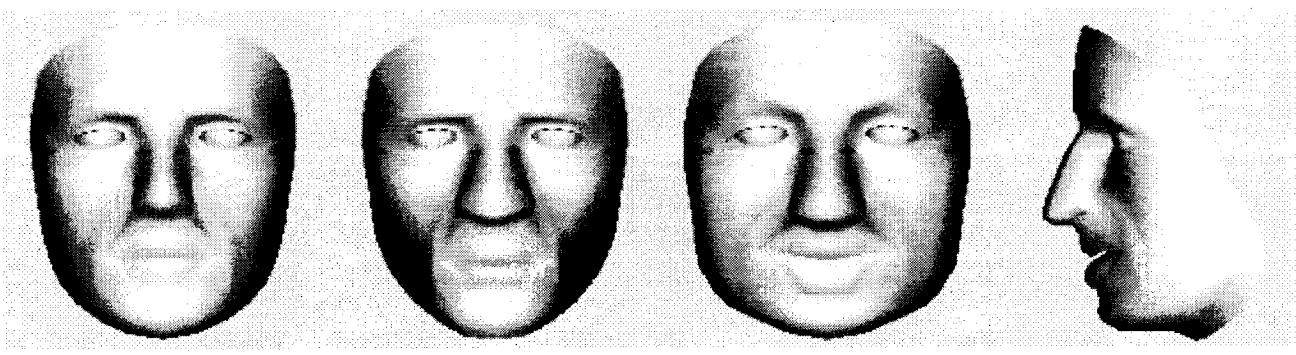


Fig. 2.10: Instances of the 3D Anthropometric-Muscle-Based-Model.

2.4. 3D AMB AAM TRAINING

The remaining of the chapter builds the proposed 3D AMB AAM. Fig. 2.11 illustrates the main components of our 3D face modeling technique. Next sections explain the training process used to derive model parameterization and fitting algorithm. Next chapter validates the obtained model in face segmentation and expression recognition experiments.

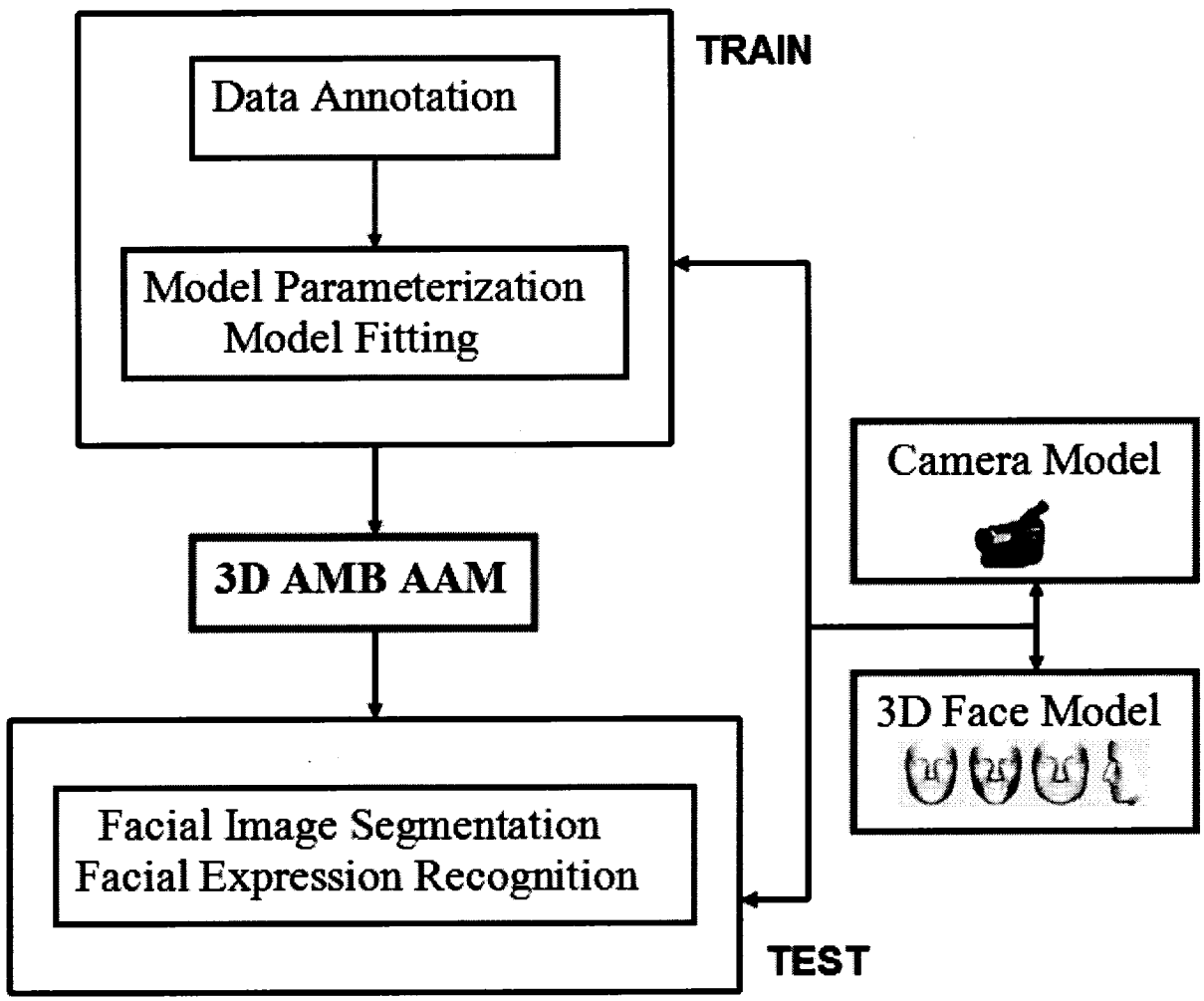


Fig. 2.11: Training/Testing the 3D AMB AAM

2.4.1. DATA ANNOTATION

First step in building an AAM is data annotation in a training dataset. We accomplish this step by using the 3D generic face, a camera projection model, and a set of training images. We intend to collect shape parameters that define a particular face and texture parameters that represent image intensities of each facial image.

Object Frame

A 3D model is build from a $3xN$ matrix, representing the concatenation of the 3D world coordinates $w_i(X_i, Y_i, Z_i)$ of all N model vertices. The model matrix can be expressed as:

$$w_i = \bar{w}_i + f_i(k), \tag{2.10}$$

where $\bar{w}_i$ is the given generic shape of the 3D model, and $f_i(k)$ represents the deformation matrix obtained by applying AAUs or/and EAUs. At any time our 3D model is defined by its pose and deformation parameters.

Camera Frame

Most model-based techniques start by assuming a perspective projection model (Fig. 2.12), often referred as a pinhole camera, which reflects the natural process of image acquisition.

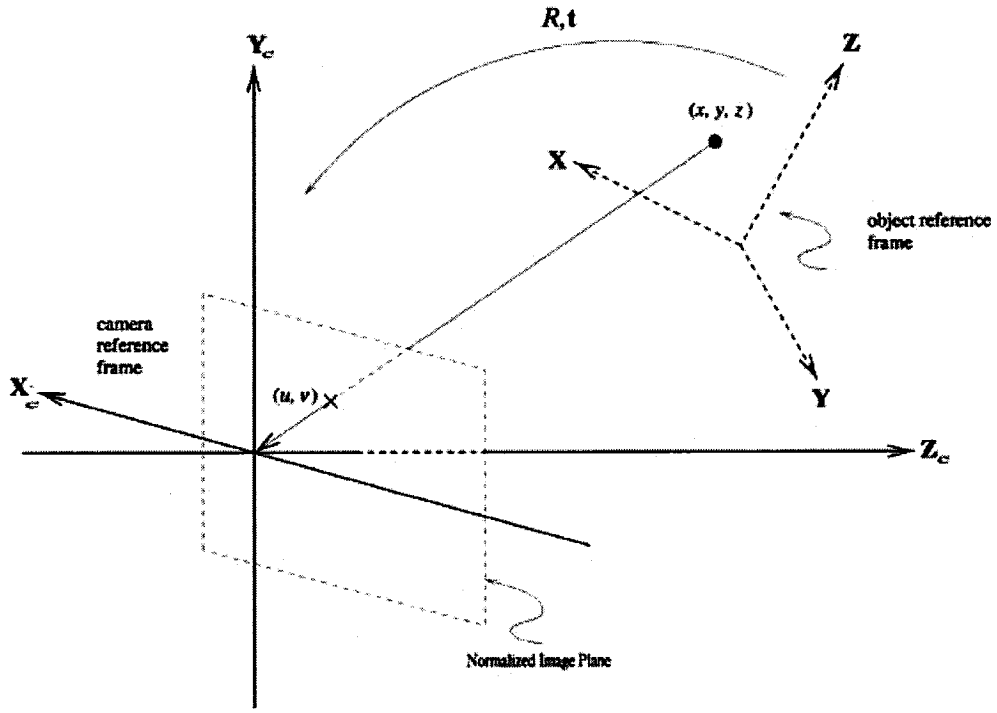


Fig. 2.12: Perspective camera model.

The 3D model points are projected in the image plane through the projection rays rooted at the center of projection (COP), located inside the physical camera. For simplicity, the origin of camera coordinate system is chosen as COP and the principal axis is traditionally aligned with the *z*-axis.

The projection of the *COP* along the *principal axis* is called the *principal point* $p_0(u_0, v_0)$. Considering $p_0(u_0, v_0)$ as the center of the image plane, and applying Thales theorem we obtain the perspective projection equation.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z^c} \begin{pmatrix} X^c \\ Y^c \end{pmatrix} \tag{2.11}$$

where $p(X^c, Y^c, Z^c)$ is a 3D object feature point represented in camera coordinate reference frame, and (u, v) is its projection onto the image plane. This simplified model taking into account

only the most important internal parameter (*focal length*) is often sufficient in machine vision application modeling.

The mapping from 3D model points to 2D image coordinates can be formalized as follows: the 3D model points $g_i(X_i, Y_i, Z_i)$ expressed in the object-centered reference frame and the same points expressed in camera reference frame $g_i^c(X_i^c, Y_i^c, Z_i^c)$ are related by a rigid transform:

$$g_i^c = Rg_i + T \tag{2.12}$$

where R, T are the rotation and translation vectors. The translation motion is modeled as a 3D position of the object reference frame relative to the current camera reference frame:

$$T = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^T, \tag{2.13}$$

with t_x, t_y components corresponding to directions parallel to image plane, and t_z the depth to the object, respectively. The rotation is modeled as a composite matrix obtained by the multiplication of the three elementary rotation-matrices accounting for the pitch, yaw, and roll of the object frame.

$$R = R_{pitch} R_{yaw} R_{roll} = \begin{pmatrix} r_x & r_y & r_z \end{pmatrix}, \tag{2.14}$$

Under the idealized pinhole imaging model with unit focal length, the object points $g_i(X_i, Y_i, Z_i)$ are projected on the normalized image plane $p_i(u_i, v_i, 1)$:

$$u_i = \frac{r_x^T g_i + t_x}{r_z^T g_i + t_z}$$

$$v_i = \frac{r_y^T g_i + t_y}{r_z^T g_i + t_z} \tag{2.15}$$

Perspective projection is a non-linear transformation that is the image measurements and their corresponding 3D object points are linked through a non-linear relation. By imposing assumptions, simplified, linear and still accurate projection models can be derived. One such example is the *scaled orthographic* or *weak perspective* projection. A perspective projection can be approximated by a weak perspective projection (i.e., linear transformation) if the object lies close to the optical axis, and the object's dimensions are small compared to its average distance Z from the camera ($Z < t_z/20$). Under the weak perspective projection, we have the following relation for each model point g_i:

$$u_i \approx \frac{r_x^T g_i + t_x}{z} \qquad (2.16)$$

$$v_i \approx \frac{r_y^T g_i + t_y}{z}$$

where z is the *scale* or *principle depth*. Usually, the principle depth is chosen as the depth of the object space origin, which is the translation t_z .

We project the 3D model onto images and adjust the AAUs and EAUs values so that the projected model best fits the facial image. We adopted a weak perspective projection model in the labeling process, since the face depth is small compared to its distance from the camera. AAUs and EAUs actuator values are recorded into a shape vector s then normalized on the interval [-1, 1] and together represent our *Anthropometric-Expression (AE) shape space*. When annotating objects in 2D or 3D using landmarks one has no knowledge about the object's pose. In this case the pose is embedded in the landmark created shape, and in order to derive the reference frame, shape alignment is necessary. The advantage of using the 3D generic model in the annotation process is that we know at each moment the pose and AE deformations. Our labeling system modifies the AE shape and 3D-pose values in small increments. After each step the texture is warped on the 3D model (Fig. 2.13).The user can check the texture mapping accuracy at any time by manipulating the 3D model. We fit the 3D model on frontal or close to frontal faces only. The fitting process starts by adjusting the 3D model pose so that the eye contours of

the model match the eyes in the image. Once the model is anchored, the AE shape is deformed for best fit model-image.

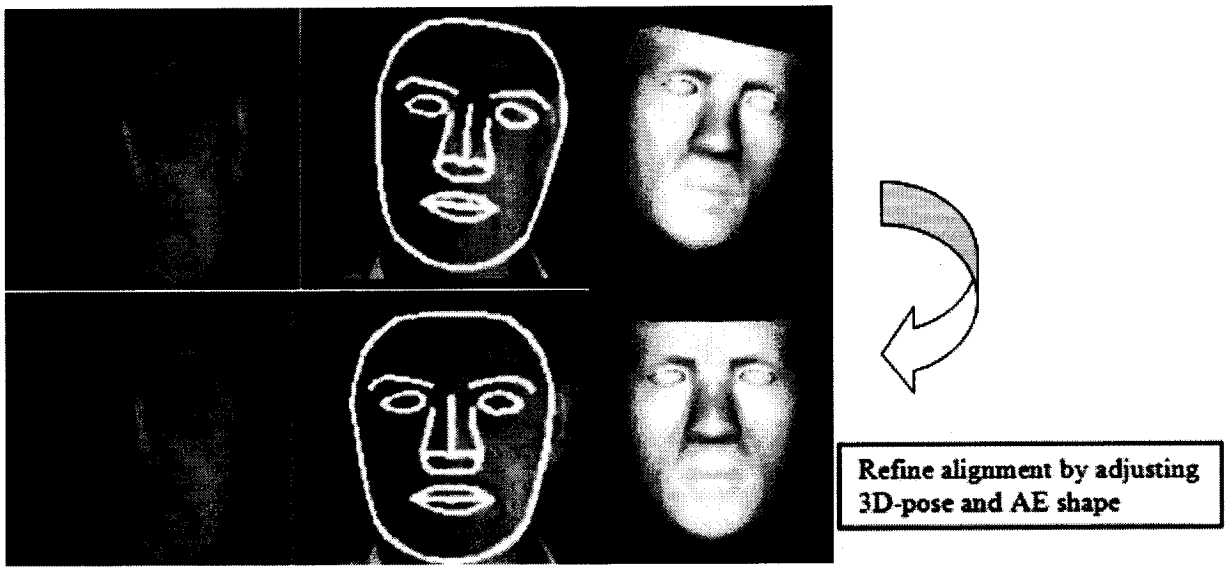


Fig. 2.13: Annotating faces using a 3D model

After the model is correctly fit on the facial image we record the facial pose: three rotations, *x-y* translations, scale $\gamma = (r_x \quad r_y \quad r_z \quad t_x \quad t_y \quad z)$, and AE deformation values. The AE actuator values are collected into a shape vector $s = (s_1 \quad s_2 \quad s_3 \quad s_4 \quad \ldots \quad s_n)$. The alignment is a simple matter of applying the inverse transform to bring the 3D face to a reference state (no pose and no animation). The recorded shape values are used to extract the texture intensities from the 3D textured model in neutral state. The training process consists in applying PCA in collected shape and texture vectors as previously explained (Section 2.2.1). This will result in two low-dimensional subspaces *AE shape space* and *appearance space* defined by the orthonormal basis U_s, and U_g respectively.

2.4.2. MODEL PARAMETERIZATION

The aim of the training is to obtain a combined compact shape-appearance representation of faces and an efficient fitting strategy.

Let N be the number of facial image training samples. Let s and g represent a synthesized AE shape and texture and let $\bar{s}$ and $\bar{g}$ represent the corresponding sample means. New instances are generated when adjusting the eigen-projections of AE shape and texture: p_s and p_g.

$$\begin{aligned} s &= \bar{s} + U_s p_s \\ g &= \bar{g} + U_g p_g \end{aligned} \tag{2.17}$$

where: U_s and U_g represent the eigen-vectors of AE shape and texture variations estimated from the training set. Since AE shape and pixel intensities use different measure units, a diagonal weighting matrix W_s is employed to compensate the differences. The W_s matrix is defined as the ratio of the AE space and texture variances using eigen-values of both models [Coo98].

In order to remove correlations between AE shape and texture the two eigen-spaces are coupled through another eigen-transform:

$$p = \begin{pmatrix} W_s p_s \\ p_g \end{pmatrix} = U_c c \tag{2.18}$$

This way we obtain a combined AE shape and texture parameterization, defined by combined c parameters. New instances of the facial shapes (physiognomies and expressions) and textures are obtained as follows:

$$\begin{aligned} s &= \bar{s} + Q_s c \\ g &= \bar{g} + Q_g c \end{aligned} \tag{2.19}$$

where: $Q_s = U_s W_s^{-1} U_{c,s}$, $Q_g = U_g U_{c,g}$, $U_c = \begin{pmatrix} U_{c,s} \\ U_{c,g} \end{pmatrix}$

Any facial shape or texture can be represented in the compressed shape-appearance space by using the projection equations:

$$c = Q_g^T (g - \bar{g})$$
$$c = Q_s^T (s - \bar{s}) \tag{2.20}$$

In order to regularize the model the eigenspaces of shape, texture and combined modes (p_s, p_g, c) are truncated, so that each represent 92% of the training set variance. To retain e percent of the variation is equivalent to keeping first M modes using the following:

$$\sum_{i=1}^{M} \lambda_i \geq \frac{e}{100} \sum \lambda_i \tag{2.21}$$

where λ_i are the eigenvalues of shape, texture and combined spaces. A high value for cutting limit results in very flexible appearance model that fits even the noise in the training images. A low e value will generate an under-trained model, incapable of fitting images of the training set. The plots in Fig. 2.14 illustrate the eigenvalues of the three spaces and the explained cumulative variance in the training set. For this training set it can be seen that 80% of the combined deformation is included within the first 13 eigenvalues and the last modes contribute little to the information. The low energy modes, usually attributed to the data noise, can safely be discarded without affecting the data set representation.

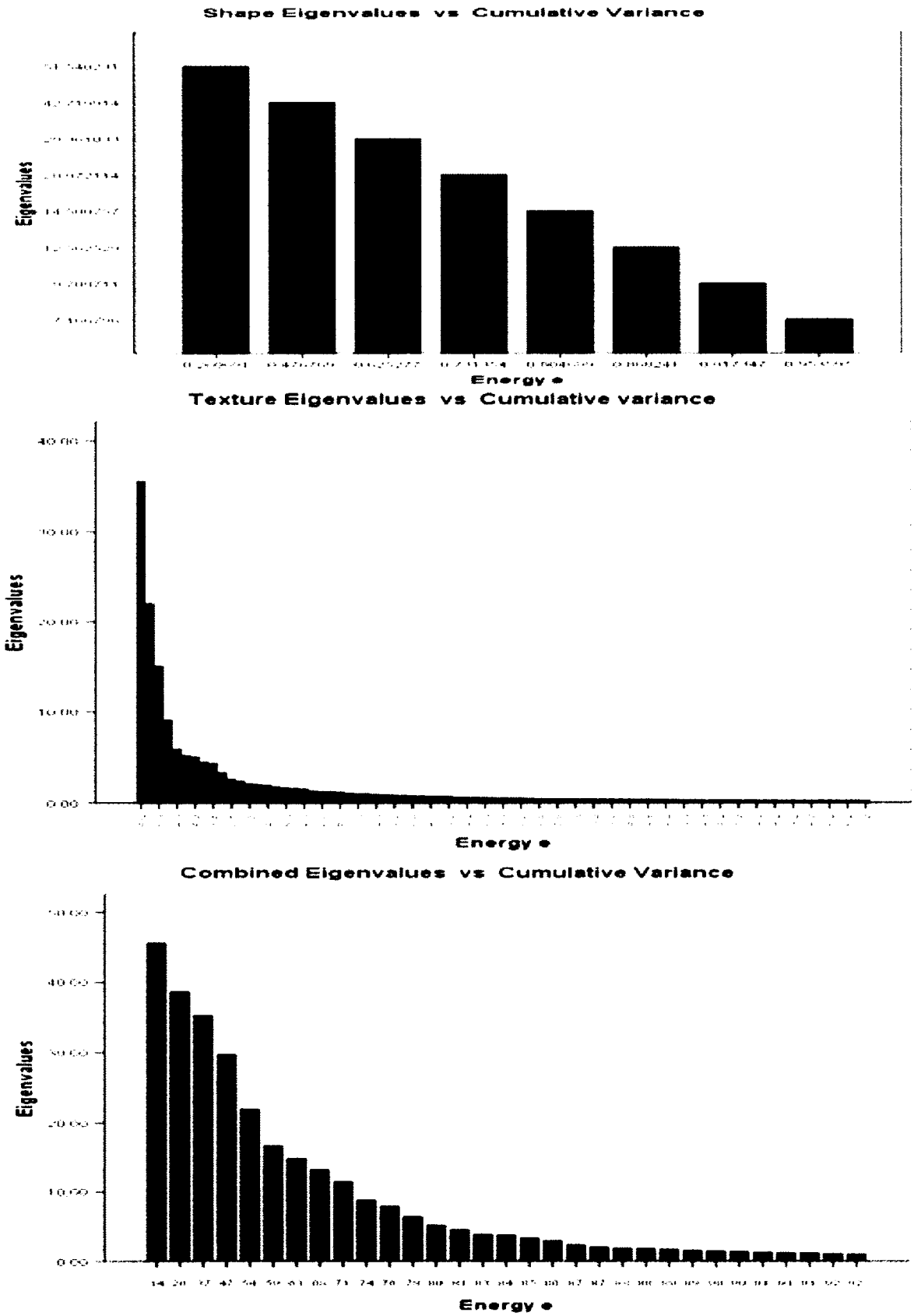


Fig. 2.14: Eigenvalues of shape, texture and combined spaces, and their contributed variance

After the last PCA step, *M-combined* modes are obtained $(M \ll N)$ representing M model parameters encoded in the vector c. One object instance (p_s, p_g) is synthesized into an image by warping the texture intensities of p_g into the geometry of the shape p_s. Fig. 2.15 shows the effects in synthesis for texture, shape, and combined modes when varying the control parameter c of the second mode $(M = 2)$ by ±2 standard deviations found in the training set.

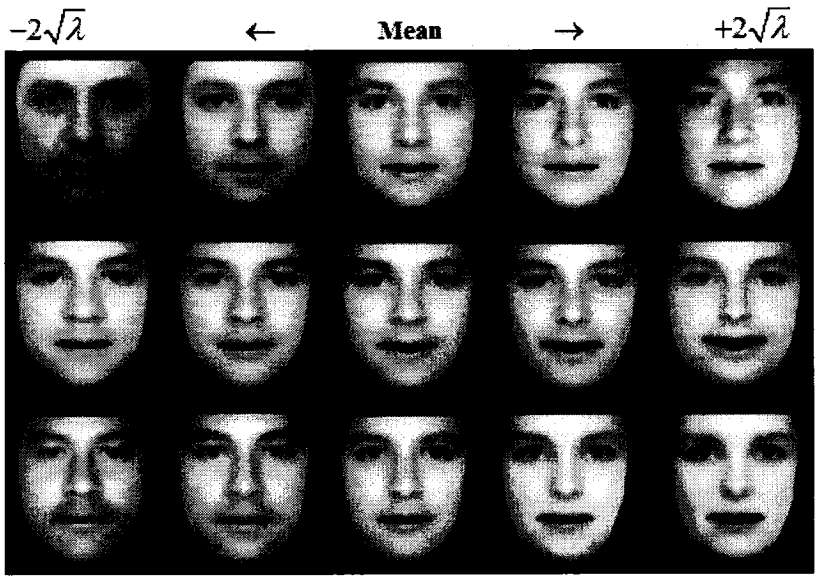


Fig. 2.15: Texture (row 1), shape (row 2), and combined synthesis (row 3) for M=2.

One important step before statistical modeling of the texture is the building of the shape-free texture [Coo98]. The shape-free framework is basically a normalization step that eliminates the geometrical differences between individuals. The normalization step will create a convex face space, which will simplify the finding of basis vectors that span the face set. The shape-free object is generated by warping the training images to the AE shape mean, computed during shape model generation. First step to synthesize the texture samples is aligning the wireframe model to the facial image using 3D transformation and model deformation. Once the alignment is completed, the texture is mapped onto the model, which is reset to the generic shape. The extracted texture from the 3D textured model is the *shape-free* facial image. Since the frontal model is described by AE deformation parameters the texture examples are mapped into and sampled from this AE shape-free reference. Fig. 2.16 illustrates the shape-free framework used

for training. The icons represent the mean face, the shape-free mask that isolates the face region and the texture variance across the training set.

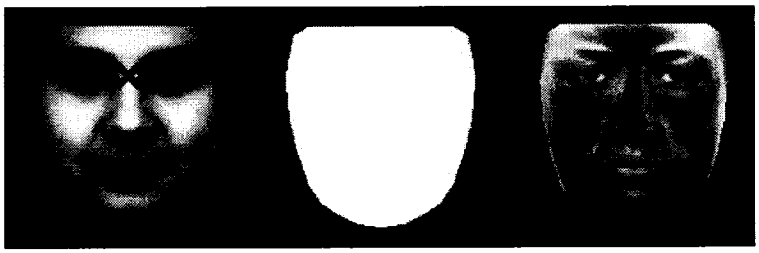


Fig. 2.16: The *shape-free* framework used for model training

2.4.3. MODEL FITTING

Each AAM comes with a search or fitting method used in image interpretation or segmentation tasks. As a model fitting strategy we adopt a simple additive update AAM scheme, which was shown to produce the best results [Coo02].

The appearance model parameters c control the shape (facial types and expressions) and texture, in the normalized model frame, according to equation (2.19). Additionally the 3D model projection in the image plane also depends on the 3D pose transformations, namely translations, rotations, and scaling $\gamma = \begin{pmatrix} r_x & r_y & r_z & t_x & t_y & z \end{pmatrix}$. That means the model appearances and deformations are defined by the parameter group (c, γ). During fitting we sample the pixels in the region of the image g_{image} defined by (c, γ), and project into the texture of the model normalized frame. The model fitting is treated as an optimization problem that minimizes the difference between the new image and the one synthesized by the 3D appearance model:

$$\partial g = g_{image} - g_{model} = g_i - g_m \tag{2.22}$$

In order to implement an efficient search algorithm we must know in advance how to correct the model parameters during image search. Basically the AAM search formulation uses a regression approach where texture difference vectors $\partial g = g_i - g_m$ correspond to model and pose parameter

displacement vectors $\partial d = (\partial c, \partial \gamma)$. The aim of the model training is to obtain an optimal prediction matrix Ψ satisfying the equation:

$$\partial d = \Psi \partial g \tag{2.23}$$

Typically, Ψ is estimated using principal component regression due to the dimensionality of the texture vectors [Coo98]. The multivariate regression scheme was then replaced by a numeric method, which is a simpler and a more robust approach [Coo99]. Let $r(d)$ be the residual vector parameterized by the model or pose parameter d to be displaced.

$$r(d) = g_i(d) - g_m(d) \tag{2.24}$$

A first order Taylor expansion of r yields:

$$r(d + \delta d) \approx r(d) + \frac{\partial r(d)}{\partial d} \delta d \tag{2.25}$$

The goal is to find δd in a way that the term $|r(d + \delta d)|^2$ is minimized. The solution of (2.25) gives:

$$\partial d = -\Psi \partial g \tag{2.26}$$

$$\text{where: } \Psi = \left(\frac{\partial r^T}{\partial d} \frac{\partial r}{\partial d} \right)^{-1} \frac{\partial r^T}{\partial d} \tag{2.27}$$

Normally during the optimization process it would be necessary to compute $\frac{\partial r}{\partial d}$ at every step, which is an expensive operation. Since AAM operates in a normalized shape-free domain, $\frac{\partial r}{\partial d}$ is considered constant for all training examples. Hence, Ψ is considered fixed and estimated off-

line during training. $\frac{\partial r}{\partial d}$ is estimated by numeric differentiation by displacing model parameters from a known value and calculating an average over the training set [Coo99]. Displacements ∂d and residuals ∂g are recorded and combined with a Gaussian kernel to smooth them. The computed Ψ matrix is subsequently used in image segmentation tasks.

The number and range of displacements should be chosen in such a way that the linearity assumption (2.26) remains true. A set of random displacements was generated by perturbing model parameters one by one for known training images. All experiments used the perturbation scheme shown in Table 2.4. Fig. 2.17 and Fig 2.18 illustrate few examples during training process, where the synthesized image is overlaid onto the original image. As in [Coo98] and [Ste00] the training process treated pose and model parameters independently generating two corresponding prediction matrices.

Variable	Displacements
Translation t_x, t_y:	$\pm 5\%, \pm 10\%$, of the width, height of the shape-free image
Rotation r_x, r_y, r_z:	± 5, ± 15 degrees
Scale z:	1.1, 0.9
Model parameters c_i:	$\pm$ 1.5, 0.5 standard deviations

Table 2.4: Displacements used in 3D AMB AAM training.

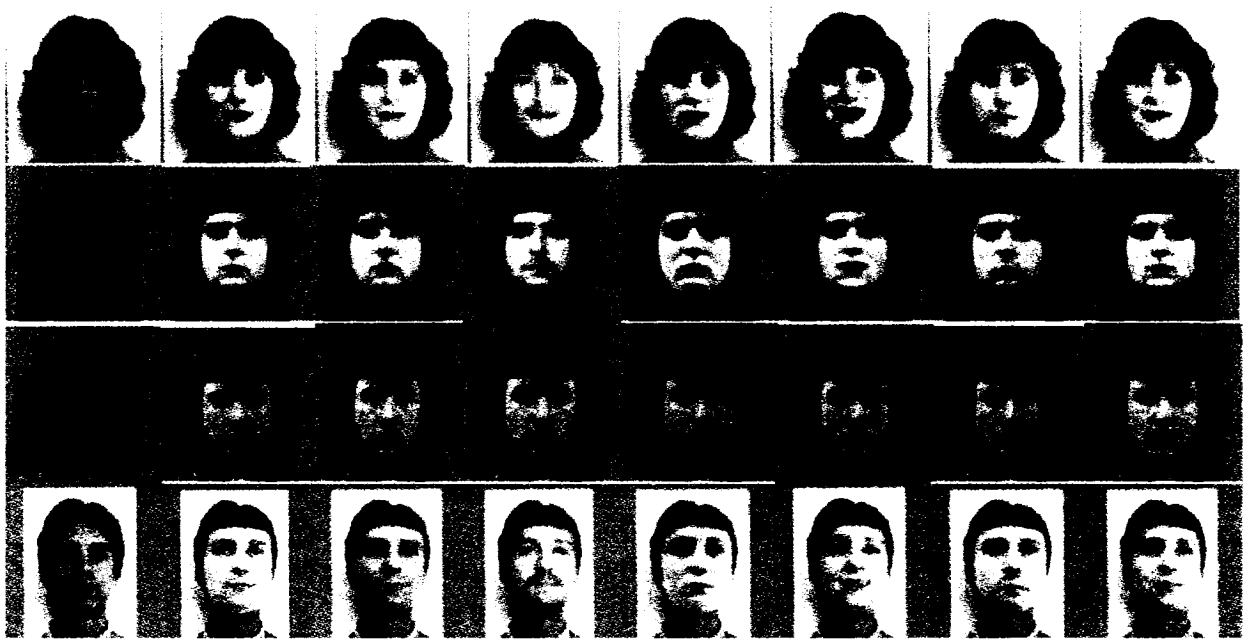


Fig. 2.17: Training for model displacements

Fig. 2.18: Training for pose displacements

The search prediction stage of the training algorithm consists of following steps:

1. Start with d_0, a known AE shape and appearance model parameters or model pose for the current image. Displace the parameters, one at the time, by a known amount ∂d.
2. Using the new values $d = \partial d + d_0$ synthesize a normalized image g_m using (2.19).

3. Compute the difference $\partial g = g_i - g_m$ and record the pair $(\partial d, \partial g)$.
4. Repeat steps 1 to 3.

2.4.4. QUALITY OF TRAINING

The effects of the displacements can be visualized using the prediction matrix using the following formulation:

$$\partial d_i = a_i \partial g \tag{2.28}$$

where: a_i is the i^{th} row of the prediction matrix Ψ, ∂d_i is the predicted change in model or pose parameters, and ∂g is the difference between sampled and synthesized image. The a_i row values represent weights corresponding to changes in the model and pose parameters $(\partial c, \partial \gamma)$ during prediction. Fig. 2.19 illustrates the weights used for the first five model parameter predictions (upper row) and the six pose parameter predictions (bottom row). Negative weights are represented by dark patches and positive weights by bright regions.

Fig. 2.19: Weights corresponding to changes in model (row 1) and pose parameters (row 2).

The quality of the training is given by three characteristics of the prediction matrix [Ste02]:

1. Ability to predict learned displacements
2. Ability to interpolate and extrapolate in between learned displacements
3. High accuracy around zero displacement

In Section 3.1., we test the ability of the model to generalize, which is fitting faces in images not seen in the training set. Firstly, we test the quality of the prediction matrix in the training set for the 3D AMB AAM and 2D AAM, which is the ability to predict learned model and pose displacements. In order to compare the quality of the two training methods we built the model in a database contains 40 frontal or near frontal (± 15 degree rotations) facial images of people exposing different physiognomies, expressions, facial hair, glasses, and illumination conditions. The 2D AAM was created using a freely available implementation [Ste00]. A total of 58 face landmarks are manually labeled on each image of the training set to be used in 2D AAM training. 3D AMB AAM used our weak-perspective projective framework to label the images. In both methods the resultant mean facial texture patches contained on average 13,000 pixels. Both models were set to retain 92 percent of the combined shape and texture variation of the training set of faces.

To test the prediction matrix properties, we displaced the face model from the true position on each image of the training set, using the values from Table 2.4, and used the model to predict the displacement given the residual error vector. The translation displacements were made with respect to the shape-free image size, namely 5% and 10% of the facial width. For our "shape-free" size these percentages translate to 5 and 10 pixels displacements. The rotation interval was chosen [-10, 10] degrees. The plots represent an evaluation of the assumption of a linear relationship between model parameters, pose parameters and the observed texture differences (equation 2.26). Fig. 2.20 shows the actual and the predicted pose displacement from a number of displacements. Table 2.5 illustrates the training ability to predict pose parameters for 2D AAM and 3D AMB AAM. The results suggest that both 3D and 2D AAM perform closely when predicting small pose displacements. This is expected since the rotation displacements are close to frontal position, maximum [-10, 10 degrees]. However, the 3D AAM has the advantage over the 2D AAM on predicting the full 6 DOF pose. We expect a difference in linearity for the c model parameters since they encode texture and shape, the main factors that provoke the model's non-linear behavior.

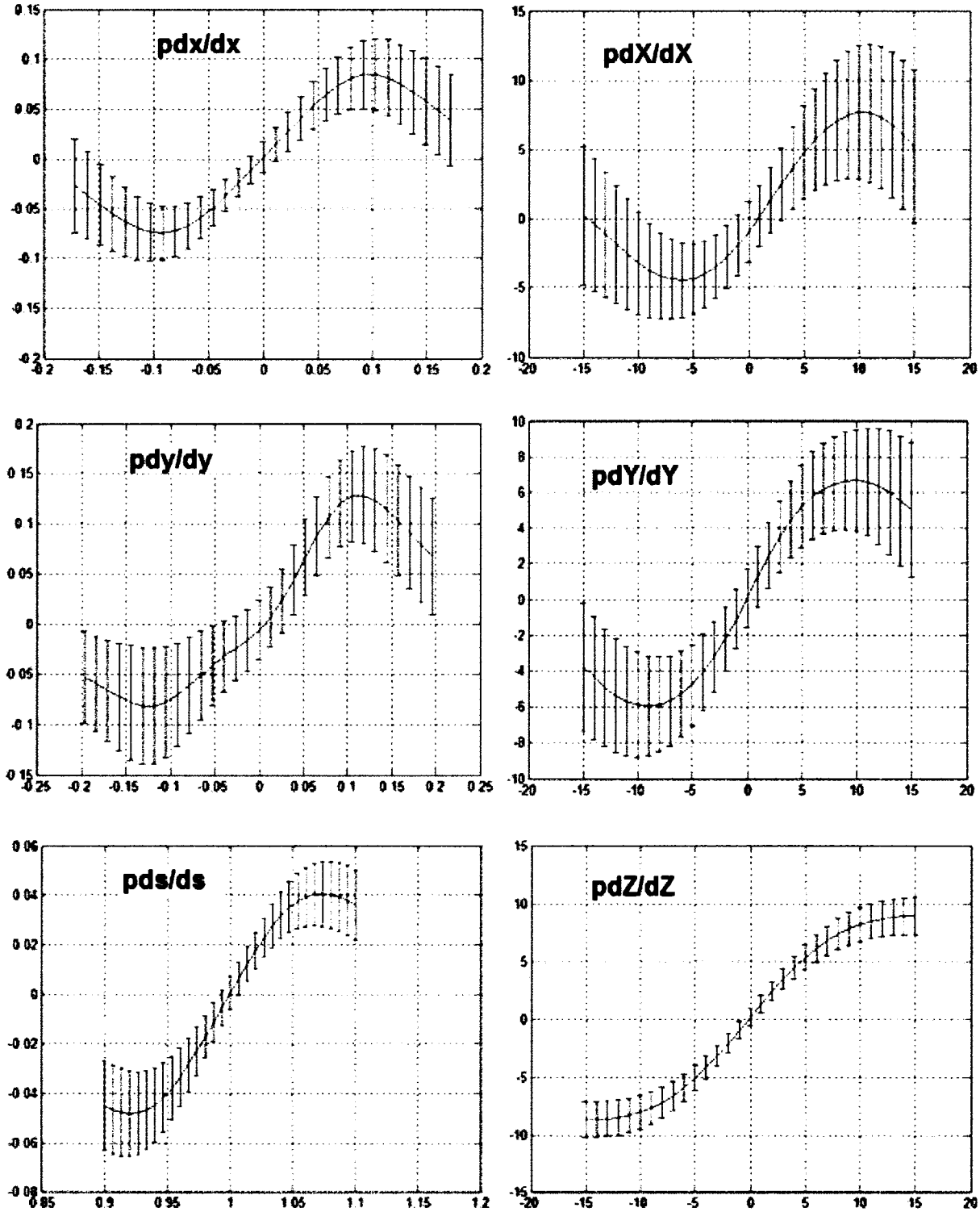


Fig. 2.20: Actual vs. predicted 3D pose displacements; translation (*x, y*), scale (*s*), and rotations (*X, Y, Z*).

// 3D, 40, 60 pixels between eyes, "shape-free"=13000 pixels		
Average X error (within +/- 10% width)	: 2.81 pixels	(2.6 %)
Average Y error (within +/- 10% height)	: 3.13 pixels	(2.5 %)
Average scale error (within +/- 10%)	:	2.09 %
Average rot X error (within +/- 10 degrees)	: 4.03 degrees	
Average rot Y error (within +/- 10 degrees)	: 2.60 degrees	
Average rot Z error (within +/- 10 degrees)	: 1.03 degrees	

// 2D, 40, 60 pixels between eyes, "shape-free"=13000 pixels		
Average X error (within +/- 10% width)	: 1.92 pixels	(1.4 %)
Average Y error (within +/- 10% height)	: 3.05 pixels	(2.3 %)
Average scale error (within +/- 10%)	:	1.21 %
Average rot error (within +/- 10 degrees)	: 0.91 degrees	

Table 2.5: Performance of 3D AMB AAM vs. 2D AAM on predicting learned pose displacements

Fig. 2.21 shows the actual and the predicted model parameter displacement for *c0*, *c2*, *c3*, and *c4*. Table 2.6 shows a comparison between the 3D AMB AMM and 2D AAM *c* parameter prediction. The maximum displacements on *c* parameters are 1.5 standard deviations, and the range [-0.7, 0.7] corresponds to less than 0.8 standard deviations (see Table 2.7). The 3D AAM behaves more linearly than 2D AAM on the range of displacements [-0.7, 0.7]. This aspect is in accordance with [Coo98], which shows that the best linearity of 2D AAM was obtained for less than 0.5 standard deviations. The results also suggest that discarding the highest variance parameters (*c0* and *c1*) from the 3D AMB AAM search process can lead to better fitting accuracy. This is equivalent with discarding the first two modes of variations (eigenvalues and eigenvectors) during model training.

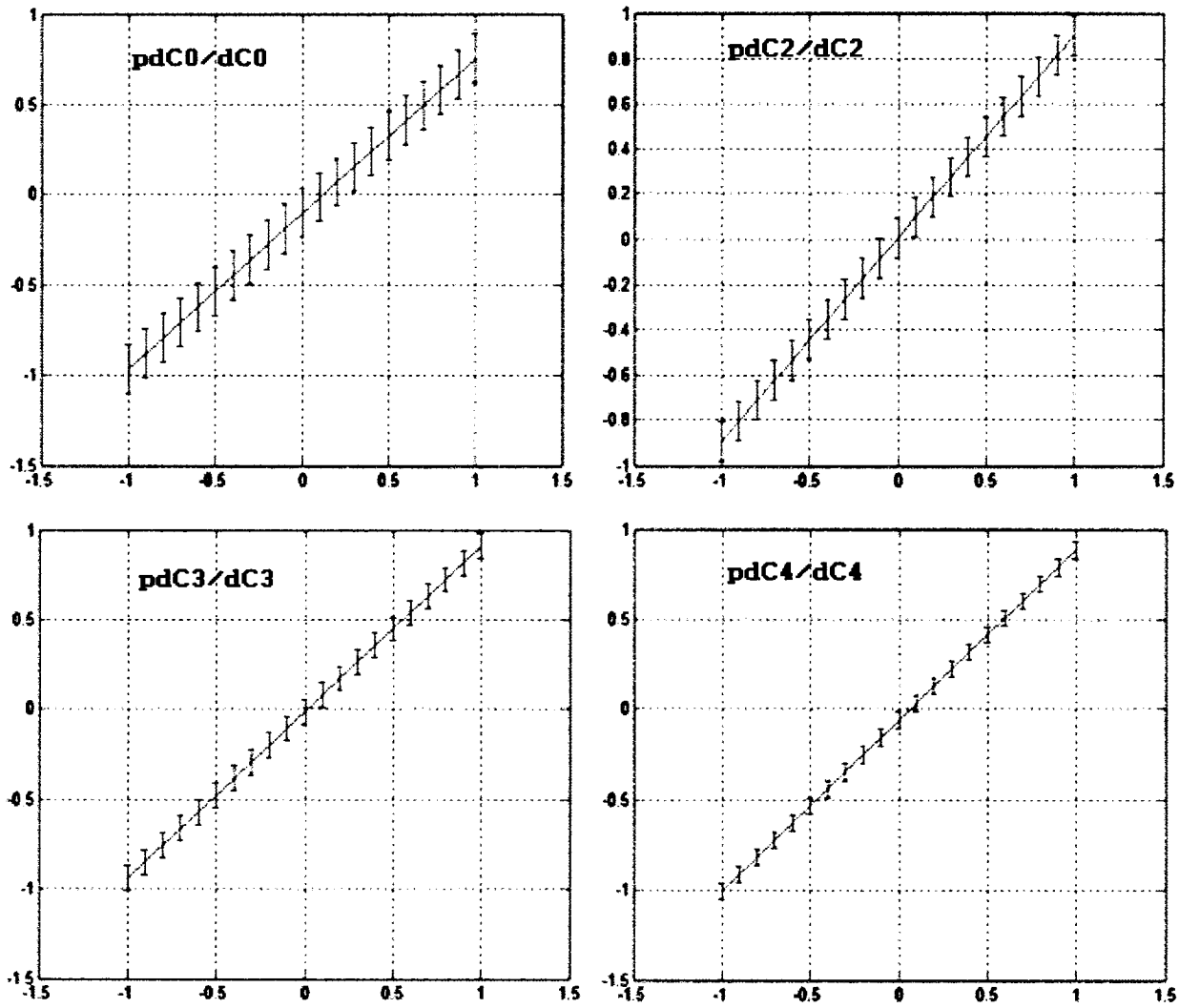


Fig. 2.21: Actual vs. predicted *c* model displacements for the 3D AMB AAM

// [-1,1], lin[-0.7,0.7], disp=1.5,0.7
Avg C0 error (within +/- 0.7) : 0.1356 (13.56 %)
Avg C1 error (within +/- 0.7) : 0.1844 (18.44 %)
Avg C2 error (within +/- 0.7) : 0.0772 (7.72 %)
Avg C3 error (within +/- 0.7) : 0.0621 (6.21 %)
Avg C4 error (within +/- 0.7) : 0.0644 (6.44 %)
Avg C5 error (within +/- 0.7) : 0.0913 (9.13 %)
Avg C6 error (within +/- 0.7) : 0.0783 (7.83 %)
Avg C7 error (within +/- 0.7) : 0.0390 (3.90 %)

// [-1,1], lin[-0.7,0.7], disp=1.5,0.7
Avg C0 error (within +/- 0.7) : 0.2082 (20.82 %)
Avg C1 error (within +/- 0.7) : 0.0648 (6.48 %)
Avg C2 error (within +/- 0.7) : 0.0998 (9.98 %)
Avg C3 error (within +/- 0.7) : 0.1200 (12.00 %)
Avg C4 error (within +/- 0.7) : 0.1985 (19.85 %)
Avg C5 error (within +/- 0.7) : 0.1953 (19.53 %)
Avg C6 error (within +/- 0.7) : 0.1370 (13.70 %)
Avg C7 error (within +/- 0.7) : 0.1711 (17.11 %)

Table 2.6: Performance of 3D AMB AAM vs. 2D AAM on predicting learned *c* model displacements

1. disp:-1.500000, std:0.605792, finaldisp: -0.908689	1. disp:-1.500000, std:0.670563, finaldisp: -1.005845
1. disp:1.500000, std:0.605792, finaldisp: 0.908689	1. disp:1.500000, std:0.670563, finaldisp: 1.005845
1. disp:-0.700000, std:0.605792, finaldisp: -0.424055	1. disp:-0.700000, std:0.670563, finaldisp: -0.469394
1. disp:0.700000, std:0.605792, finaldisp: 0.424055	1. disp:0.700000, std:0.670563, finaldisp: 0.469394
2. disp:-1.500000, std:0.584359, finaldisp: -0.876538	2. disp:-1.500000, std:0.599944, finaldisp: -0.899916
2. disp:1.500000, std:0.584359, finaldisp: 0.876538	2. disp:1.500000, std:0.599944, finaldisp: 0.899916
2. disp:-0.700000, std:0.584359, finaldisp: -0.409051	2. disp:-0.700000, std:0.599944, finaldisp: -0.419961
2. disp:0.700000, std:0.584359, finaldisp: 0.409051	2. disp:0.700000, std:0.599944, finaldisp: 0.419961
3. disp:-1.500000, std:0.567273, finaldisp: -0.850909	3. disp:-1.500000, std:0.569444, finaldisp: -0.854167
3. disp:1.500000, std:0.567273, finaldisp: 0.850909	3. disp:1.500000, std:0.569444, finaldisp: 0.854167
3. disp:-0.700000, std:0.567273, finaldisp: -0.397091	3. disp:-0.700000, std:0.569444, finaldisp: -0.398611
3. disp:0.700000, std:0.567273, finaldisp: 0.397091	3. disp:0.700000, std:0.569444, finaldisp: 0.398611
4. disp:-1.500000, std:0.519854, finaldisp: -0.779781	4. disp:-1.500000, std:0.555484, finaldisp: -0.833226
4. disp:1.500000, std:0.519854, finaldisp: 0.779781	4. disp:1.500000, std:0.555484, finaldisp: 0.833226
4. disp:-0.700000, std:0.519854, finaldisp: -0.363898	4. disp:-0.700000, std:0.555484, finaldisp: -0.388839
4. disp:0.700000, std:0.519854, finaldisp: 0.363898	4. disp:0.700000, std:0.555484, finaldisp: 0.388839

Table 2.7: Displacements, standard deviations, and final displacement for the first four *c* model parameters in the training set

The obtained results show comparable accuracies of the 3D AMB AAM and 2D AAM on predicting learned 2D pose displacements on small intervals. The results also show that 3D AMB AAM behaves more linearly than the 2D AAM on specified intervals for *c* model displacements. Cootes et al. showed [Coo98] that the linear region of the curves extends over a larger range at the coarser resolutions. This confirms the idea that building multi-resolution AAM can help the fitting accuracy, besides improving the speed.

In order to asses the capability of the 3D AMB AAM in predicting orientation outside the bounds of [-10, 10] degrees, we chose the following rotation displacements (degrees) about X, Y Z axes: -27,27,-20,20,-13,13,-7,7,-3,3. We observe (Fig. 2.22) the model acts linearly on the interval:

1. [-20,20] in predicting yaw angle
2. [-15,15] in predicting roll angle
3. [-30,10] in predicting pitch angle

The graph for predicting the pitch angle appears shifted due to the unsymmetrical face appearance relative to X-axis.

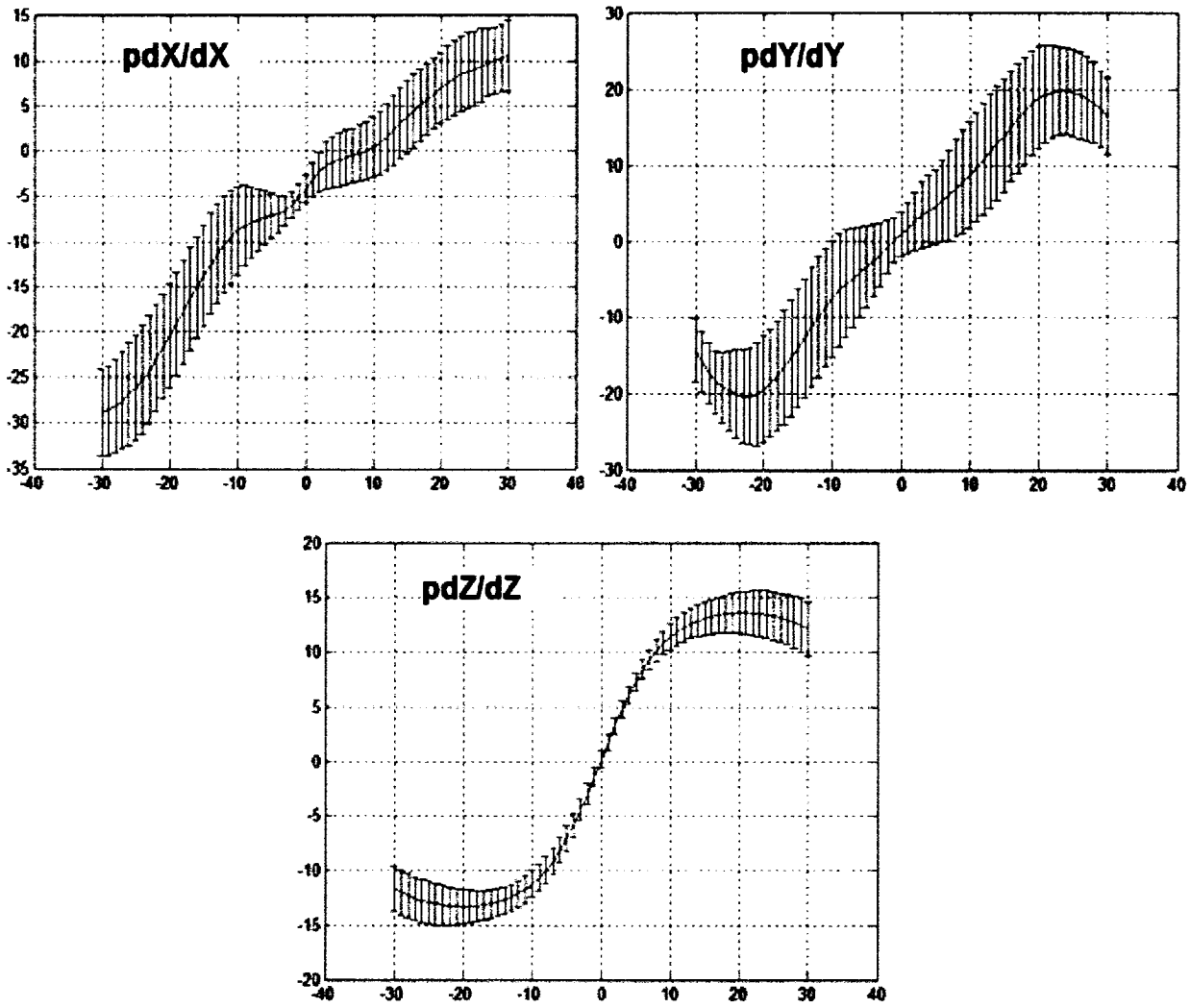


Fig. 2.22: Extending the interval of predicted pose displacements for 3D rotations around *X, Y, Z* axes

2.4.5. THE ILLUMINATION PROBLEM

Illumination changes are one of the major causes of poor fitting. Compensation of light variation could be of great benefit for model-based image segmentation tasks. Existing light normalization algorithms can be divided in two categories:

1. Model-based: an explicit model describes illumination changes induced by varying the light source.
2. Image Processing: image enhancement algorithms, such as gamma correction or histogram equalization.

Model-based Light Normalization

An explicit light model requires training images of the same subject, acquired under different lighting conditions. One idea is to use PCA to represent lighting conditions as a low-dimensional subspace of the image space. Recently, Gross et al. [Gro04] introduced the eigen-light fields, which is a PCA texture model built using light fields. They employed the model in a pose-invariant face recognition algorithm. Light field AAM solves limitations induced by current 2D or 3D AAM. They can model non-lambertian surfaces of complex scenes and pose variation. Christoudias et al. [Chr04] built a 3D AAM from 2D facial appearances and light fields, and use it for facial image segmentation tasks. Varying illumination on image sequences was estimated without light source and reflection models using only linear techniques in [Eps95], and [Ram02].

Using a 3D model for face analysis has the advantage of explicitly including the photometric effects in the algorithm. The 3D surface normals allow for estimating the illumination parameters using linear [Eis97] or non-linear [Sta95] optimization. Eisert [Eis02] builds eigen-light maps of explicit 3D models. Diverse light scenarios are covered using blending between different light maps. Explicitly modeling the light conditions in the 3D AMB AAM remains an important area of our future work. The present work employs a photometric normalization of facial images.

Image Processing Light Normalization

Image processing methods attempt to bring all facial images to a canonical illumination. Basically, in one step, the original images are transformed into better representations with "identical" lighting conditions for the task at hand. The method has the main advantage over model-based techniques in that there is no training phase and it is independent of the fitting algorithm.

The 3D AMB AAM employs photometric normalization as a preprocessing step of training and testing phases. The chosen technique is composed of two algorithms: retinex [Lan71], [Lan86], [Hor74] followed by histogram equalization.

Retinex, whose name is derived from the words retina and cortex is a theory originally proposed by Land and McCann [Lan71]. The algorithm models the lightness and color perception of the human vision. Since the original proposal in 1971 retinex theory has experienced changes [McC76], [Fra83], [Lan86], [Mar00], and has been applied to various image enhancement tasks such as dynamic range compression [Fun99], color correction [Fun99], [Mar00] or shadow removal [Fin02]. The retinex algorithm calculates the lightness at each image pixel in the log domain by taking in consideration the interactions between pixels along a chosen path. The path of the comparison pixels confers the main difference between the retinex algorithms. Finally, histogram equalization creates a uniform distribution of the image brightness. This step brings training and test images processed by retinex to the same dynamic range of intensity. Fig 2.23 illustrates the effect of applying the light correction on images exposing arbitrary illumination conditions.

Fig. 2.23: Light normalization by image processing techniques (images from BU image sequence database [Sca98]).

Basic photometric normalization is also part of AAM construction. There are normally two simple correction methods applied to the training and test images:

1. Zero Mean: the mean of the image set is removed from each image (equation 2.3).

2. Zero Mean and Unit Variance: the mean of the image is removed and the pixel values scaled by their standard deviation.

Chapter 3

Face Modeling Validation

We validate our 3D AMB AAM model in two types of experiments performed on static images:

1. Facial Image Segmentation
2. Facial Expression Recognition

The image segmentation procedure is a low-level, model fitting technique. We compare the performance of the 3D AMB AAM with 2D AAM in facial image segmentation tasks. The classical 2D AAM is build on frontal or near frontal faces exposing different expressions and lighting conditions. The findings are simply based on the distance of the fitted model points from the ground-truth ones, using no high-level semantic knowledge about the face (pose, expression etc.). The second experiment expresses the ground truth in high-level semantic terms, namely labeled facial expressions. The experiment evaluates the 3D AMB AAM model's capability in recovering facial expressions by comparing it with various existing methods.

3.1. FACIAL IMAGE SEGMENTATION

In this section we compare 3D AMB AAM with 2D AAM and show that 3D AMB AAM fitting has a better accuracy on static facial images. Fitting the 3D model to a facial image is a time consuming search in 2N dimensional space (every vertex has 2 coordinates in the image). In order to increase the speed and accuracy of fitting, we choose the AE space as search space instead of the 2D shape space. The search space is further reduced when searching in AE eigen-space, hence allowing only a small set of eigen-deformations. In order to fit the model to a facial image we employ the iterative updating scheme based on a principal component regression [Coo98]. Matching optimization is performed iteratively. At each iteration, the model predicts the changes in pose and model parameters that lead to a better model to image fit. Convergence is

declared when the error is below a chosen threshold. The searching algorithm consists of following steps:

1. Start with d_0, an initial AE shape and appearance model parameters and model pose for the current image, and compute the image difference $\partial g = g_i - g_m$ and its error.
2. Extract the corresponding correction ∂d (2.26).
3. Set $k = 1$
4. Compute new model parameters $d - k\partial d$, the corresponding difference ∂g and its error.
5. If *error* < *threshold* accept the parameters and go to step 2.
6. Else try different settings for k such as 1.5, 0.5, and repeat steps 4 and 5 until reaching a maximum number of iterations or the error is below the desired threshold.

For this experiment we built a person independent model, which means the persons from the testing set were not included in the training set. Most of today's AAM methods have been tested only on the set of images used for training, or different images containing persons used in training set. In order to compare the accuracy of the two algorithms, 3D AMB AAM and 2D AAM, we use two image datasets, the training and the testing set. The training database contains 270 frontal or near frontal (± 15 degree rotations) facial images of people with age ranging from teen to old age, exposing different shapes, expressions, and illumination conditions. The 2D AAM was created using a freely available implementation [Ste00]. A total of 58 face landmarks are manually labeled on each image of the training set to be used in 2D AAM training. 3D AMB AAM used our projective framework described in previous chapter to label the images. In both methods the resultant mean facial texture patches contained on average 14,000 pixels.

There are factors representing a trade-off between segmentation speed and reconstruction quality, such as:

1. The number of texture and shape modes. The reconstruction and generalization accuracy increase with the number of modes, and the speed decreases since the computation of the analysis-synthesis step grows linearly with the number of modes.
2. The size of shape-free or reference texture. A larger texture size finds better facial features, but slows down the search algorithm. Since we are testing the accuracy of the method in static images, we employ a normalized image of resolution 256 by 256 pixels.

The facial patch occupies 20% of the normalized frame size. In a head pose tracking scenario [Dor03] a normalized face patch of 40 × 42 pixels might be large enough to reconstruct faces in enough detail to track its motion.

We tested the 2D and 3D AAM on 124 ground-truth annotated facial images from the *Olivetti Research Laboratory's* (ORL) database [Oli98]. The faces in the test set were not present in the training set. The image search experiment makes the assumptions that each image contains only one face, and faces are captured by a fixed camera under weak perspective projection. The 3D texture warping is achieved using available OpenGL graphics hardware. In our tests we assumed known eye positions. The searching 2D and 3D models were resized by scaling them from 0.9 to 1.1.

Two sets of experiments were performed. The first test evaluates the model fitting starting with a "correct" initialization (left model eye position matched image eye position). In the second experiment we make the search problem "harder" by displacing the initial model from the correct position by 25% of the distance between eyes. Both methods converge in about 10 iterations.

We compare 3D AMB AAM with 2D AAM by evaluating three different properties of an ideal AAM [Sty03], [Dav02]:

1. Compactness: the ability to describe the model with a minimal set of parameters.
2. Generalization: the model ability to generate instances unseen in the training set.
3. Specificity: the model ability to generate only valid instances of the class.

Each criterion uses the distance to manually selected anatomical landmarks as goodness of fit measure. Even with the help of human knowledge it is impossible to establish a perfect correspondence of all landmarks across the training set. The most difficult to register consistently would be the interpolated landmarks, which do not represent anatomic entities such as eye corners, or the nose tip or image evidence entities such as points of high curvature. The comparative evaluation used a small set of anatomical landmarks as ground-truth points, manually selected by a human expert on each face image. We computed the mean square error ε between the annotated and corresponding reconstructed landmarks L for each method.

$$\varepsilon(M) = \frac{1}{L}\sum_{i=1}^{L} |u_i - u_i'(M)|^2 + |v_i - v_i'(M)|^2 \quad (3.1)$$

where u_i, v_i represent the annotated landmarks, u_i', v_i' the reconstructed landmarks projected from the 3D model using (2.18) and (2.15), and M is the number of combined shape-appearance modes. In our tests we use the maximum number of modes for each method.

3.1.1. COMPACTNESS

A compact model is characterized by small variance and parameter number. The 2D AAM model is described by 31 shape modes, 106 texture modes and 53 combined modes that retain 92 percent of the combined shape and texture variation of the training set of faces. The 3D AMB AAM model is described by 9 shape modes, 81 texture modes and 33 combined modes that retain 92 percent of the combined shape and texture variation of the training set of faces. The 3D AMB AAM trained model appears more compact than the 2D AAM.

3.1.2. GENERALIZATION

The model generalization is computed by running the two algorithms in the test set, and is defined by the averaged approximation error $G(M)$.

$$G(M) = \frac{1}{N}\sum_{i=1}^{N} \varepsilon_i(M) \quad (3.2)$$

where ε_i represents the distance between the found facial shape to the ground-truth one in the test set, and N is the number of annotated facial points.

The experiments illustrated in Fig. 3.1 (a, b) have shown that the 3D AMB AAM fitting results are more accurate than those of 2D AAM (Table 3.1). During the two experiments ("correct" and "harder" initialization) performed on 124 images the 3D AMB AAM was more accurate in 103, respectively 106 cases.

Variables	2D-AAM generalization error	3D-AMB-AAM generalization error	3D-.vs.2D-AAM performance
Experiment-exact-initialization	89.57	49.60	83.0%
Experiment-displaced-initialization	145.63	66.65	85.4%

Table 3.1: Search accuracy of 3D AMB AAM compared to 2D AAM.

Fig. 3.2 (a, b, c, d, e, and f) illustrates the 3D AMB AAM search on images from the test set. The initial hypothesis for face location and search results at different iterations is overlaid onto the original image. The final iteration illustrates the model reconstruction and contours of the found facial shape. Fig. 3.3 (a, b, and c) shows the superior 3D AMB AAM segmentation performance illustrating the convergence processes of 3D AMB AAM vs. 2D AAM. Facial images in Fig. 3.1, 3.2, and 3.3 are from ORL database [Oli98].

3.1.3. SPECIFICITY

A specific model should generate only "legal" instances of the object class. The specificity is measured by randomly generating a set of S model instances, and comparing it to the training set. The model instances are obtained by varying the c parameter in (2.19) within the range found in the training set. The measure used for comparison is the averaged approximation error $S(M)$.

$$S(M) = \frac{1}{S}\sum_{i=1}^{S} \varepsilon_i(M) \tag{3.3}$$

where ε_i represents here the distance between the generated shape to the nearest one in the training set. We generated $S = 210$ face instances in this experiment. The 3D AMB AAM (6.28 specification error) is more specific than the 2D AAM (8.66 specification error).

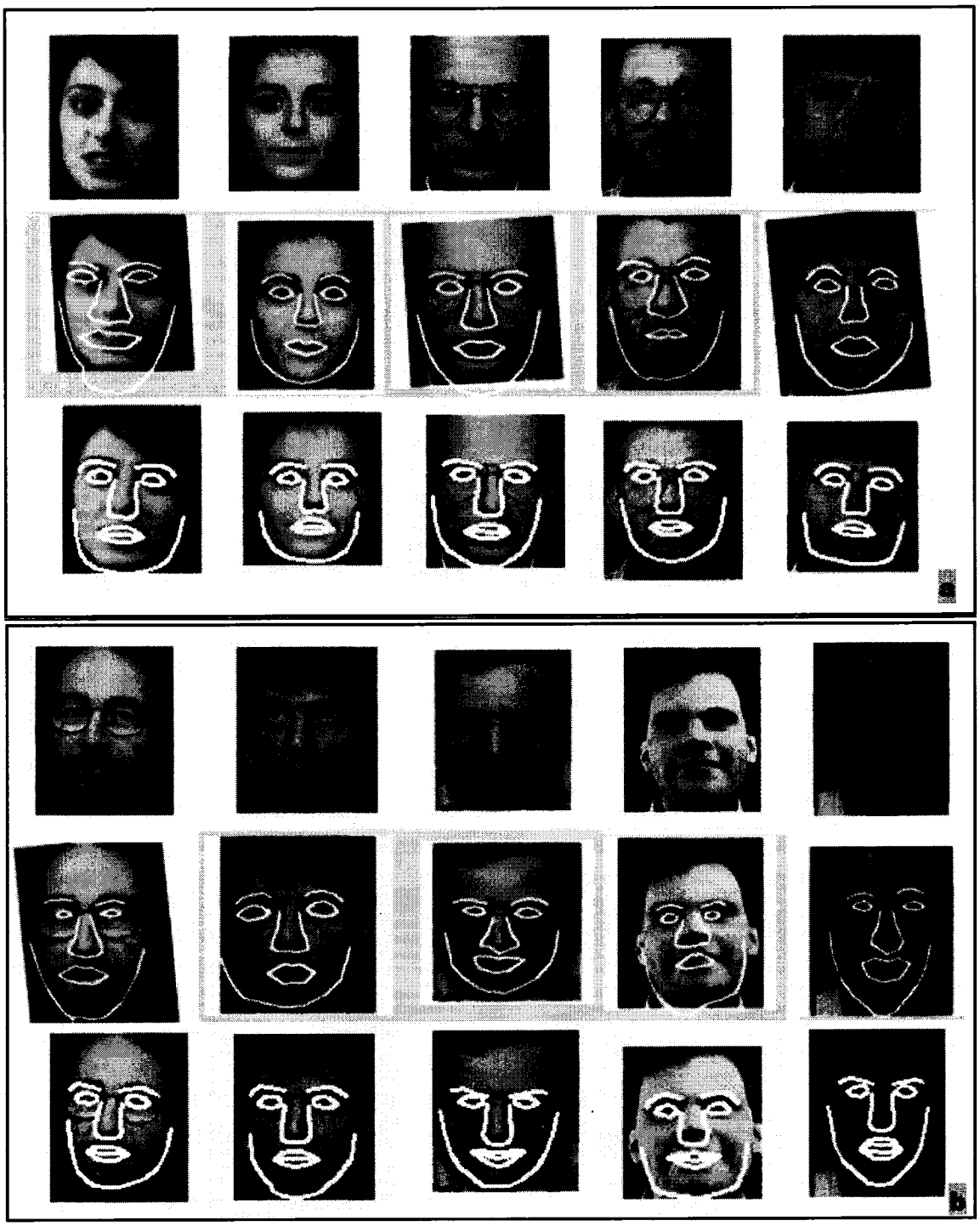
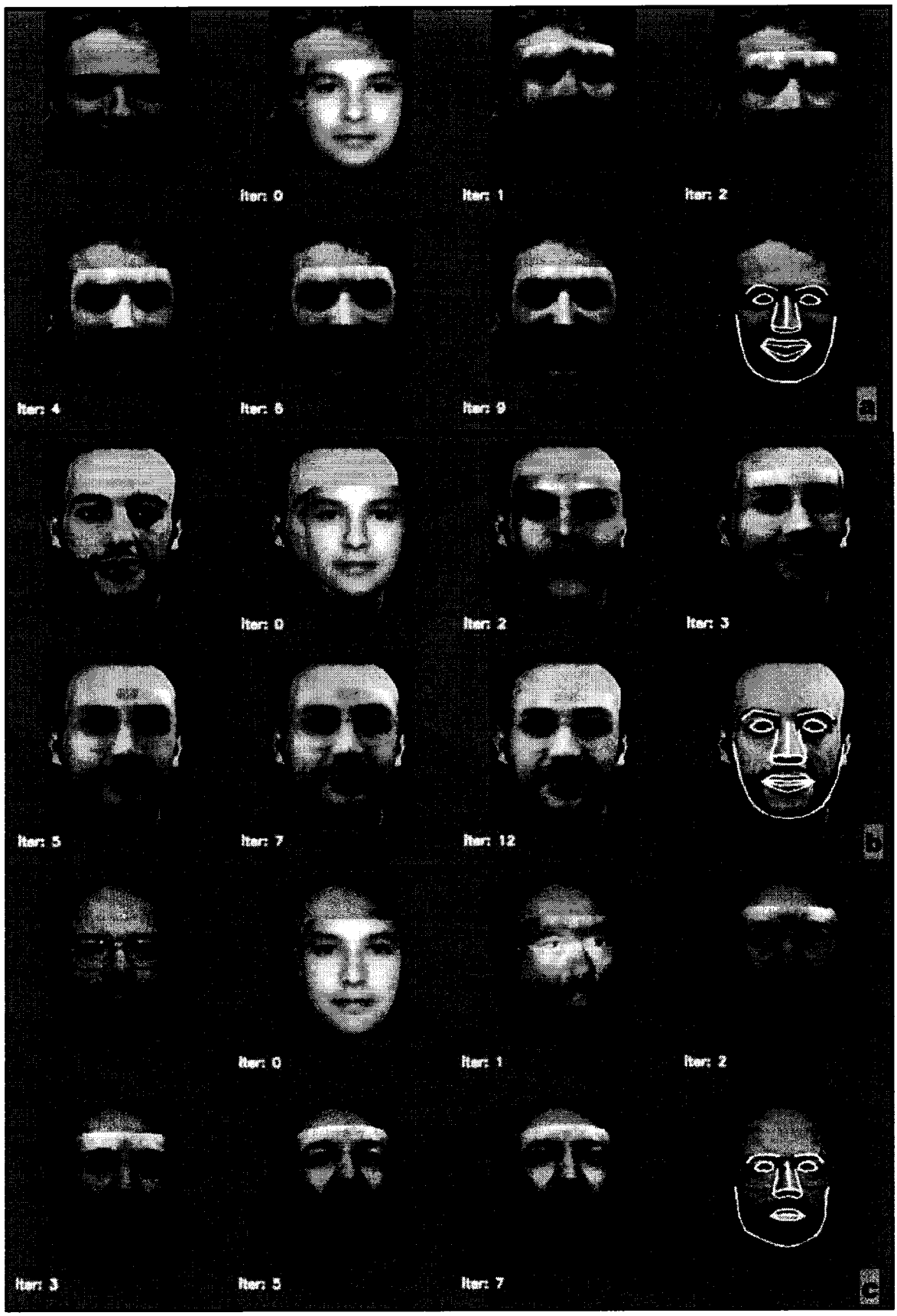


Fig. 3.1: Generalization test (a, b) of 2D AAM (second row) and 3D AMB AAM (third row)


Iter: 0
Iter: 1
Iter: 2
Iter: 4
Iter: 6
Iter: 9
a
Iter: 0
Iter: 2
Iter: 3
Iter: 5
Iter: 7
Iter: 12
b
Iter: 0
Iter: 1
Iter: 2
Iter: 3
Iter: 5
Iter: 7
c

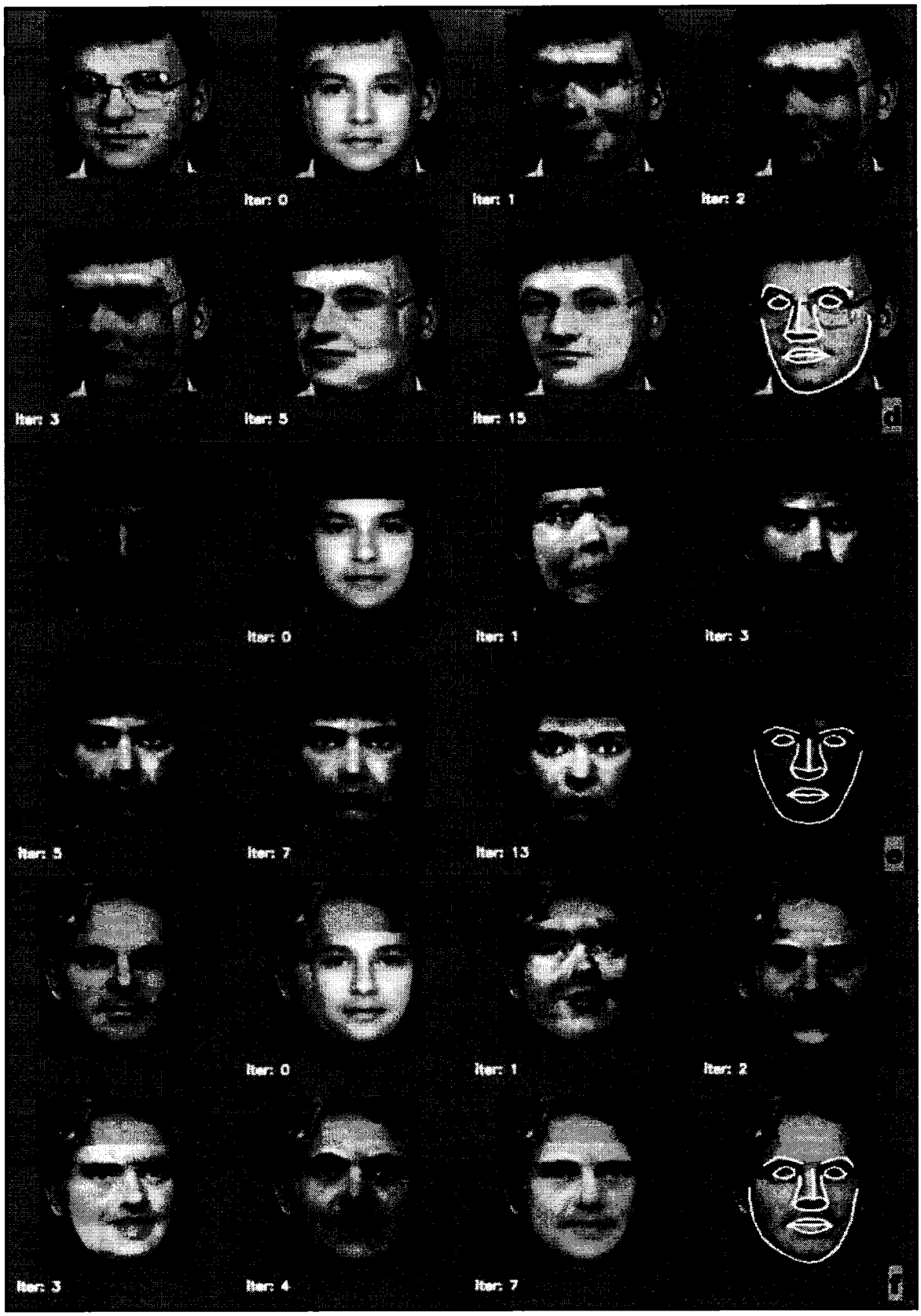
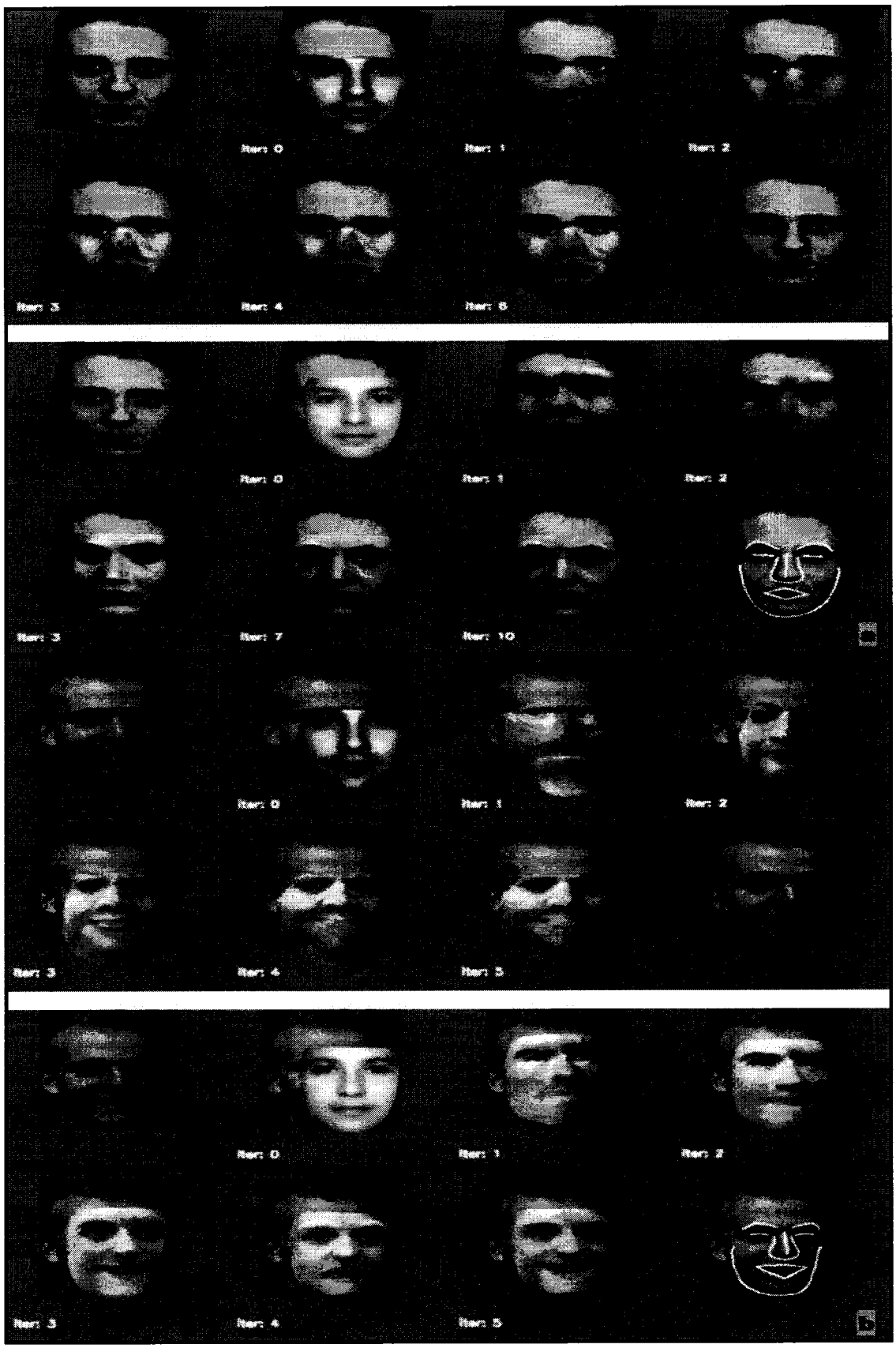


Fig. 3.2: The 3D AMB AAM facial image search with initial displacement (a, b, c, d, e, f).



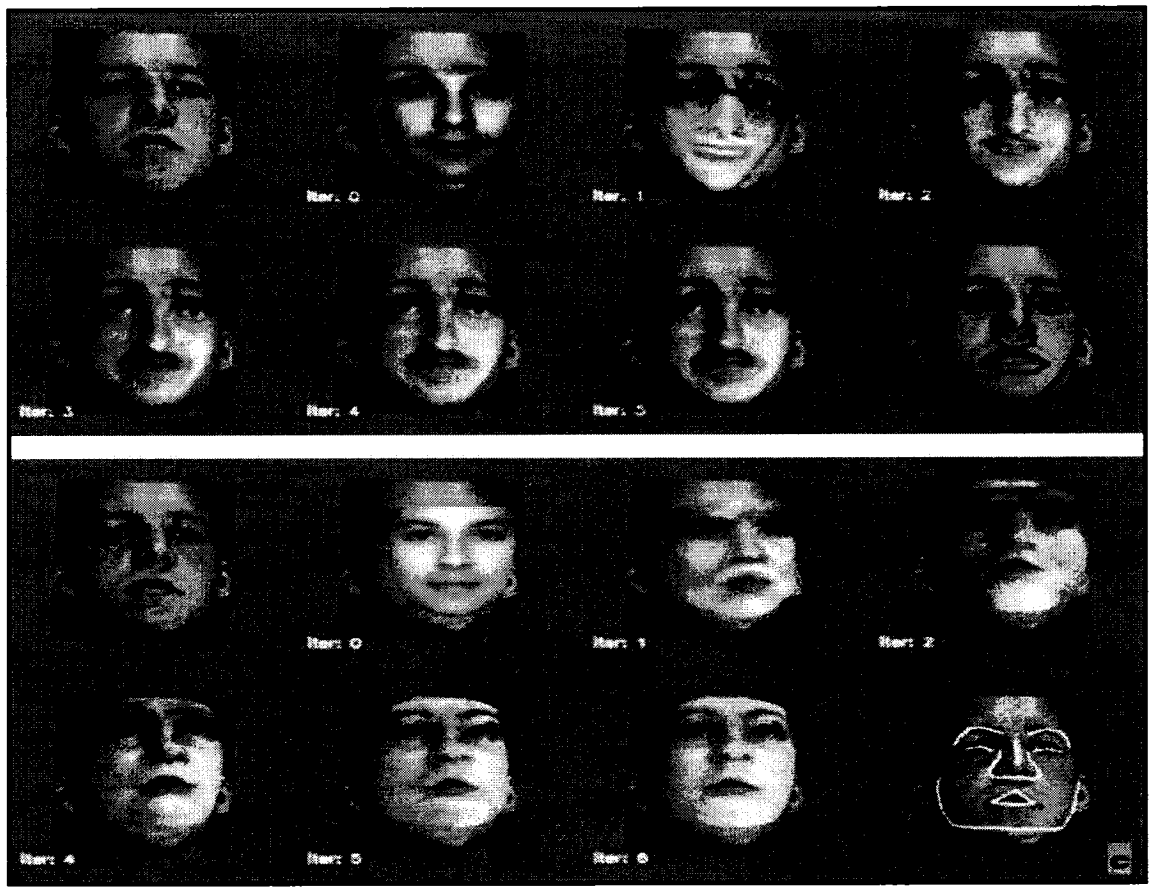


Fig. 3.3: The 3D AMB AAM (second group) vs. 2D AAM (first group) convergence accuracy (a, b, c).

3.2. FACIAL EXPRESSION RECOGNITION

The problem of facial expression recognition includes multiple areas. An ideal facial expression analysis system is automatic, real-time and has to deliver accurate results on all these areas.

3.2.1. STATE OF THE ART

The characteristics of an ideal facial expression analysis system are presented in Table 3.2 [Pan00b], [Tia03]. The real-time characteristics apply for systems that deal with sequences of images.

Ideal Facial Expression Analyzer Properties	
Generality	Handle large head motions No restriction on subject age, ethnicity or appearance Handle light variation Handle occluded faces No usage of artificial markers Recognize all facial expressions
Automatic	Automatic face acquisition Automatic facial expression data extraction Automatic facial expression classification
Real-time	Real-time face acquisition Real-time facial expression data extraction Real-time facial expression classification

Table 3.2: Properties of an ideal face expression analyzer [Pan00b].

Surveys [Fas03], [Pan00b] described the work in automated facial expression analysis in images and video. Any facial expression analyzer has three modules [Tia03], [Fas03], [Pan00b]:

1. Face acquisition
2. Feature Extraction
3. Expression Classification

Face acquisition should automatically locate the face area in the input image or sequence. In the case of static images the process is referred to as *detecting* a face. In the case of sequence of images the process is referred to as *tracking* a face. Once the face is found, the facial area is further processed to extract facial features.

Facial features can be extracted from facial images using three types of methods [Fas03]:

1. Analytic or Feature-based, [Pan00a], [Coh98]
2. Holistic or Model-based, [Edw98b], [Hon98], [Bla97], [Hua97]
3. Hybrid or Analytic-to-Holistic, [Yon97], [Lyo99], [Ess97], [Wan98], [Zha98]

Feature-based methods localize the facial features of an analytic face model in the input image or track them in the image sequence. Model-based methods fit a holistic face model to the face in the input image or track it in the image sequence. Deformable templates in general and active models (ASM, AAM) [Coo95] in particular represent good examples of the holistic approach to extract facial features.

Facial expressions can be classified in terms of facial actions that cause an expression (AUs), or in terms of emotional (also called prototypical or basic) expressions (happiness, surprise, anger, sadness, fear, and disgust). Each emotional expression is composed as a set of AUs. The majority of methods on expression analysis perform an emotional classification. There are several issues related to the expression classification [Pan00b]. First of all the widely used FACS rely on Ekman's linguistic description of the six basic facial expressions. There is no unique description based on universal facial codes or actions. Secondly, each expression depends on the subject's physiognomy and the timing and intensity of facial deformation. These factors make automated expression recognition a difficult task. The most used classification techniques are [Pan00b]:

1. Model-based, [Edw98b], [Lyo99], [Hon98], [Ess97], [Hua97]
2. Neural-Networks, [Yon97], [Zha98]
3. Rule-based, [Pan00a], [Bla97].

A facial expression analyzer is also:

1. Static: classifies expressions in a single frame [Edw98b], [Pan00a], [Lyo99], [Hon98], [Yon97], [Hua97], [Zha98]
2. Dynamic: classifies expressions in image sequences using temporal information [Bla97], [Coh98], [Ess97], [Wan98].

Hong et al. [Hon98] employ a holistic method to find facial expressions in static images. They model the face using a labeled graph with vertices called jets. Each jet is a Gabor wavelet with five frequencies and eight orientations that locally extracts image information. They perform a coarse-to-fine image search to best fit the labeled graph to the facial geometry [Wis95]. In order to classify facial images into one of the six basic plus neutral emotion categories, Hong et al. splits the training database in galleries of expressions. Each fitted graph is compared against all

expression galleries and attributed to the closest one. The recognition rate was 89% on the training set and 73% in case of novel subjects.

Yoneyama et al. [Yon97] use a hybrid approach to face representation in static images. They compute the optical flow between a neutral face and facial expression images. For each person in the training set a vector of 80 facial motion parameters is extracted and used as an input for two 14x14 Hopfield neural networks (NN). They attempted to recognize four basic expressions (surprise, anger, happiness, and sadness). The average success rate was 92%. However, their method was tested in the training set, deals only with vertical facial movement, and doesn't handle facial hair, glasses or rigid head motion.

Lyons et al. [Lyo99] uses a hybrid approach to represent the face. They use a graph with 34 jets labeled in 256x256 pixel images, to classify expressions into the six basic plus neutral emotion categories. They collected the expression vectors and separated them in clusters of facial attributes using Linear Discriminant Analysis (LDA). Six trained binary classifiers and a nearest-neighbor classifier decides the correct expression of the input vector. They achieved a recognition rate of 75% on new subjects. Zhang et al. [Zha99] use a similar approach as in [Lyo99] to extract the facial expression data. They employ a 680x7x7 NN for classification, achieving a recognition rate of 90.1% in the training set.

Huang et al [Hua97] employ a holistic method to find facial expressions in static images. They build a PDM from 90 facial points manually labeled on 90 images of 15 Chinese subjects, exposing the basic six expressions. They fit the PDM to the face geometry by applying gradient-descent shape parameter estimation. In order to classify the basic six expressions, they define "Action Parameters" representing differences between the fitted PDM and the labeled facial shape. The achieved a correct recognition rate of 84.5% on a testing set containing persons from the training set.

Pantic et al. [Pan00a] use a feature-based model to represent the frontal and the side view of the face in static images. The chosen points correspond to the facial features and specific shapes as chin or mouth. Using multiple detectors and anatomical rules about the feature constellation they fit the model to faces. They classify the facial deformations in AU-classes using production rules

based on FACS. Pantic et al. tested their algorithm in 496 dual-view images and reported a recognition rate of 92% for upper face AUs, and 86% for lower face AUs. The second test compared the found AUs to the AUs of the six basic emotional expressions. The achieved recognition rate was 91% in 256 dual-view images of eight subjects. However, their system can not handle light variation, or faces with hair or glasses. The system is also slow; the processing speed takes 8 seconds per image on a Pentium 2.0 GHz.

Black and Yacoob [Bla97] built local parameterized models of image motion to analyze facial expressions in image sequences. They use three flow models: affine, planar, and affine-plus-curvature. The planar model deals with the rigid motion and the affine models represent deformations of the face. The extracted motion parameters are used to form rules for classifying the six basic expressions. Their algorithm achieved a recognition rate of 88% in a database of 70 image sequences containing 40 persons exposing 145 expressions.

Essa and Pentland [Ess97] employ a hybrid approach to face representation in image sequences. They represent the rigid and non-rigid motion by spatial-temporal motion-energy templates. The face and features are detected and tracked using an eigenspace technique [Pen94]. Their dynamic model is based on the optical flow computation between two normalized faces. To regularize the motion, a multi-scale coarse-to-fine Kalman filter is applied. As a classifier, they used Euclidian norm of the distance between learned motion templates and the observed ones. The test was performed in 52 frontal face image sequences exposing six expressions, with a success rate of 98%.

Edwards et al. [Edw98] used a holistic approach, namely a 2D AAM to extract the facial expression data in static images. They employ LDA to separate interclass variability (identity) from the intraclass variability (expressions). The training set of 200 images exposed limited light variation and pose changes. They achieved a recognition rate of 74% for the six basic expressions shown by 25 subjects. The success rate was low considering that the testing set was the same as the training set.

Wen et al. [Wen03] proposed a holistic method to track expression under various lighting conditions, and skin color. They start by fitting a 3D face model to facial image in the first frame

using anatomical landmarks as anchor points. The interframe motion is extracted using optical flow. Their expression tracking system is not automated and does not handle large head motions.

Cohen et al. [Coh03] proposed a system that recognizes 4 prototypic expressions of 5 people in a video stream. A 3D wireframe is fit onto a face and a Tree-Augmented-Naive Bayes classifier decides the expression class. They obtained an average expression recognition rate of 76.5%.

Bartlett et al. [Bar99] proposed a hybrid system composed of optical flow with feature analysis and a 61x10x6 feed-forward NN to recognize 6 AUs from frontal facial images. They achieved 91% accuracy. More recently, Bartlett et al. [Bar04] built an automatic system achieving a 95% recognition rate of 18 AUs in frontal-view face image sequences. For feature extraction and classification they used Gabor filters and Support Vector Machines (SVM).

Lien et al. [Lie98] proposed a hybrid system to detect individual and combinations of facial AUs in image sequences. They extract facial expression data using feature point and dense flow tracking with PCA. They use Hidden Markov Models (HMM) for classification, achieving an accuracy of 85% for feature point tracking, and 93% for dense flow tracking.

Automatic FACS-based expression classification remains a difficult problem to solve. Most of today's systems try to recognize individual AUs or the six (or less) prototypic expressions. Our system was tested in static images in two types of experiments. Firstly, we performed person dependent experiments, in which each subject's data is split into training and testing sets. Secondly, we performed person independent experiments, in which the testing data contained only novel persons, unseen in the training set. For the static based method each image was labeled to one or combinations of AUs. The accuracy is measured with respect to the classification output of each image.

The 3D AMB AAM is a model-based facial expression recognition technique. In our case, feature extraction and expression classification represent one model-based entity due to the semantic level of our labeling and training process (AE space). When the model is fit to a facial image, the final match provides the pose and expression of the searched face. This is an advantage over other facial expression analyzers that use different components to extract features

and classify expressions. For example, the 2D AAM fits a facial image in terms of contour deformations without using the expressions that caused them. Hence, it needs to train a classifier to learn the correspondence deformations-expressions.

3.2.2. EXPERIMENTS

The classical 2D AAM recovers the point positions of the 2D shape without the knowledge of the underlying facial types or expressions. The resulting shape points are then used for classification tasks. The main advantage of formulating the 3D shape model in AE space is that the anthropometric characteristics and facial expressions are obtained directly during the synthesis phase.

Databases

One of the problems in all emotion recognition research from static and sequences of images is the lack of testbed databases. One such database recently built is Cohn-Kanade [Kan00] database. This database contains over 2000 facial images of people exposing different facial expression. The image sequences have been FACS coded by certified experts. We received the database after our validation work completed. Testing our system in this database remains a part of our future research. The present experiment relies on two other databases recently made available for benchmarking expression recognition algorithms.

The first database used in our experiments is the Facial Expression and Emotion Database of the FG-NET consortium (Face and Gesture Recognition Network) [FGnet]. It contains image sequences to assist research on human facial expressions (Fig. 3.4). The main characteristics of the database are:

1. allowance of spontaneous and natural expressions
2. no restrictions in head motion
3. good lighting conditions, constant background

The database contains images collected from 19 individuals, each performing all six basic expressions and the neutral sequence three times. This way everyone recorded 21 sequences,

which totals 399 sequences in the database. Each sequence starts with a neutral expression, it develops without following any pattern, namely it can increase or decrease in intensity at any time, then ending again with a neutral face.

The second database used in our research is the MMI (Man Machine Interaction) Facial Expression Database [Pan05]. It provides a large test-bed for research on automated facial expression analysis, and was already used in Pantic's [Pan00a] method validation. The database is composed of more than 1500 static images and image sequences of faces displaying the prototypic expressions, as well as single and multiple AU-based expressions (Fig. 3.4). The database includes 19 different persons, ranging in age from 19 to 62, with European, Asian, or South American ethnic background. The main characteristics of the database are:

1. 79 staged emotional and AU expressions in frontal and profile views.
2. allowance of minimal head motion
3. good lighting conditions, cluttered background

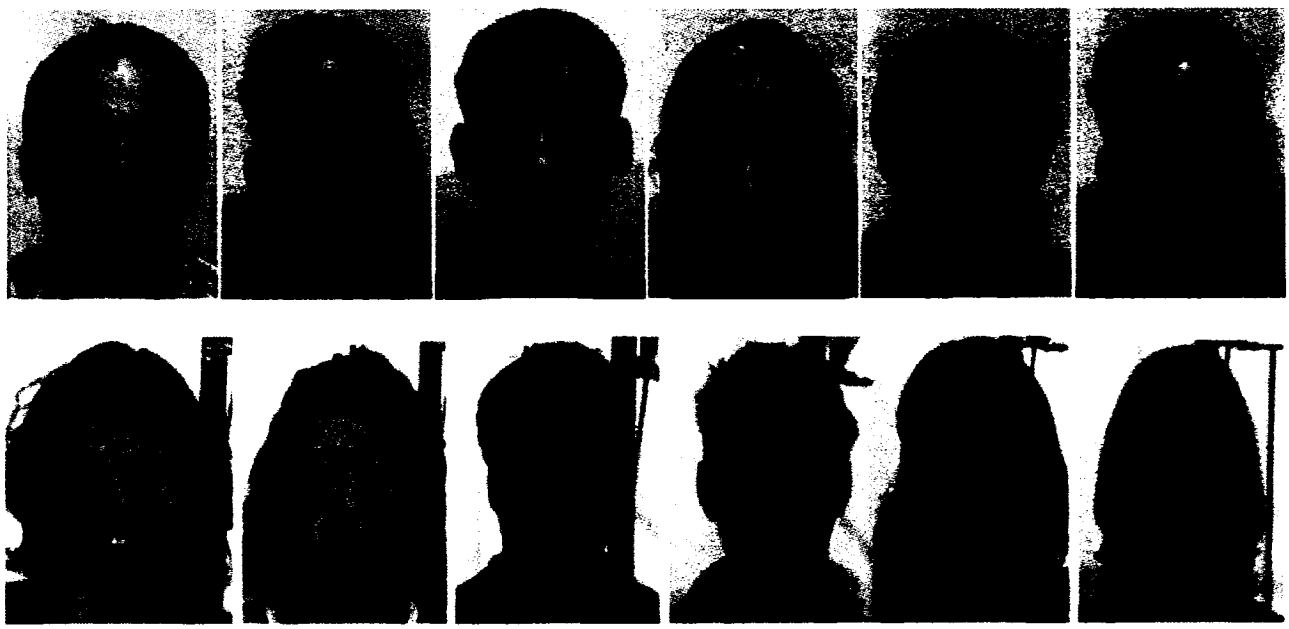


Fig. 3.4: Examples of emotional facial expressions from MMI and FG-NET databases.

For *face segmentation* model fitting criteria was expressed in terms of global fitting of the entire face. For *facial expression recognition* we use local fitting of the main facial features such as eyes, nose and mouth. The "shape-free" framework used for face segmentation may not be

suitable for facial expression or emotion recognition. The full facial image contains information that is irrelevant to the task at hand. This could affect the ability of a correct classification. In recognizing facial expressions, structural differences in faces need to be removed or minimized. The way to do this is to cut out regions of the facial image where variation provides no information related to expressions. Our perception of emotions is based on local changes in facial muscles around the eyes and mouth [Ekm77]. Under these circumstances we used a different "shape-free" framework that emphasizes the facial regions responsible for producing expressions. This way, the model fitting dependency on facial edges and background is removed. Fig. 3.5 illustrates the *inside-face* framework used for training and validation in the task of facial expression recovery. The icons in Fig. 3.5 represent the full mean face used in segmentation, the new *inside-face* mask, the extracted mean of the texture-of-interest and the appearance variance across the training set.

Fig. 3.5: The *shape-free* framework used for facial expression recognition.

The aim of this experiment is to recognize the expressions of the subject not to estimate the identity parameters encoded in anthropometric parameters. Still, we allow for a minimal variation of anthropometric parameters in order to compensate for missing facial instances from the training set.

In facial expression recognition experiments we represent the model shape with the same AAU set used in face segmentation experiments. The chosen EAUs (Section 2.3.1) describe the facial expression space. The EAUs were constraint to follow expression signatures on the interval [-1, 1]. During the labeling process we minimized the AAUs variation to fit the training image, allowing for a greater EAU variation. Some of the labeling instances are illustrated in Fig 3.6. The 3D AMB AAM was trained using a similar perturbation scheme to the one employed in face

segmentation experiments (Table 2.4). During the validation stage the trained model was required to fit the facial image using the strategy described in Section 2.3. This way, a set of EAUs model parameters was extracted for each face, which was used for classification experiments.

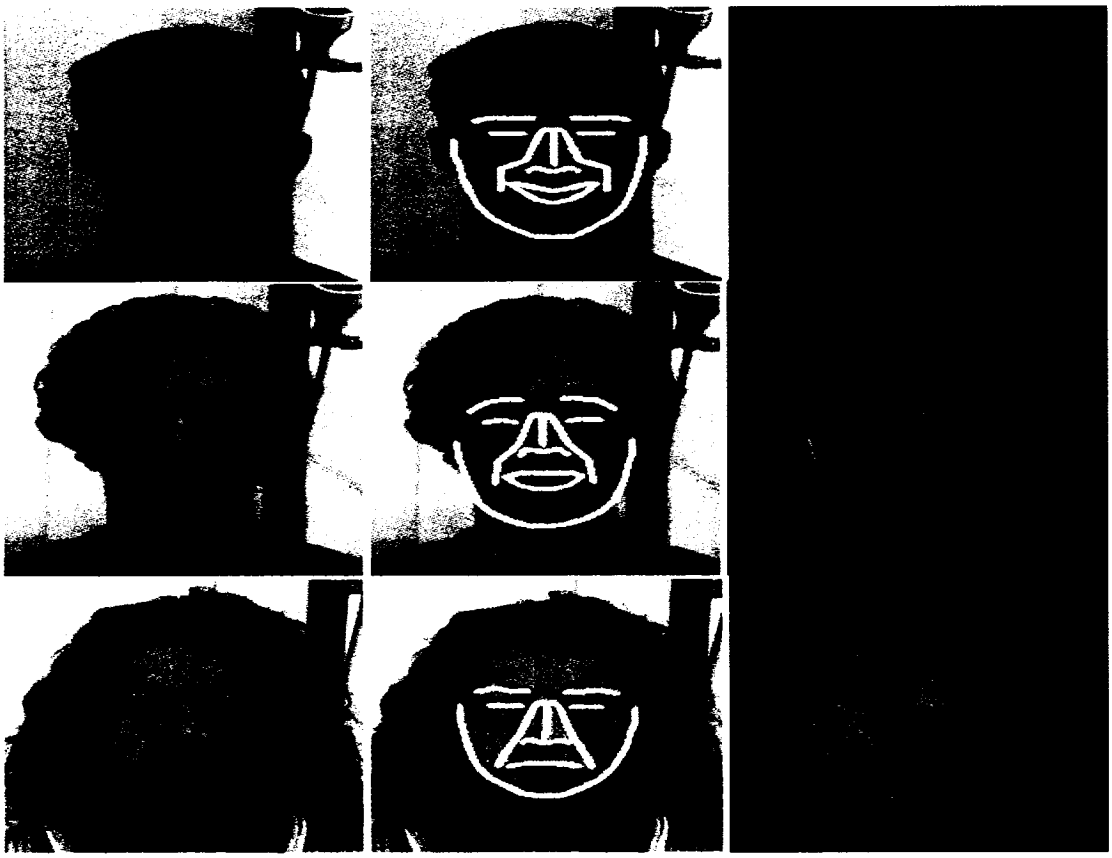


Fig.3.6: A typical labeling sequence for "happiness", "surprise", and "sadness" expressions

Fig. 3.7 illustrates instances of the model and pose perturbation scheme used for training. The four icons represent:

1. the perturbation synthesized instance overlaid onto the original image
2. the sampled shape-free image
3. the synthesized image in the new shape-free framework
4. the image difference between sampled and reconstructed instance

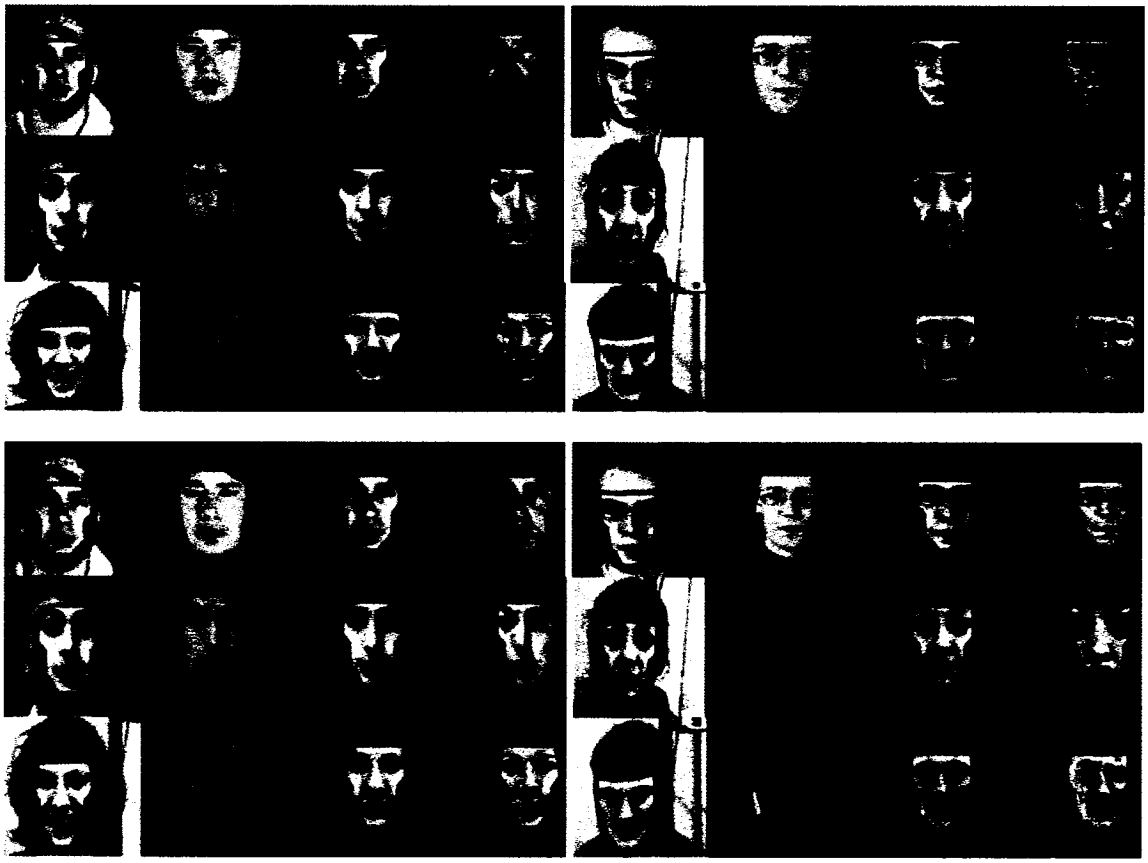


Fig. 3.7 Instances of model and pose perturbation scheme used for training.

The 3D AMB AAM was trained in individual frames extracted from MMI and FG-NET database sequences. Due to its unconstrained nature FG-NET database contains a high number of blended-expression sequences. We selected sequences with distinguishable prototypic expressions from 10 persons. We used images from the first two sequences of a person for training. The third sequence was used to perform person-depended experiments. For each person we extracted two frames for training from the first sequences, namely one neutral frame and one peak frame. A similar approach was used to extract the training frames from MMI database. This way we collected a number of 250 training images.

This time, the trained 3D AMB AAM model is described by 12 shape modes, 88 texture modes and 55 combined modes that retain 97 percent of the combined shape and texture variation of the training set of faces. The normalized training images had a resolution of 256 by 256 pixels, containing a facial texture patch of approximately 12,000 pixels. The images are pre-normalized prior to processing to ensure that frames have exact geometric correspondence. The normalized

framework was defined by the landmarks *n* (*nasion,* skull feature between the eyebrows) and *sn* (*subnasale*, the center of the nose base), which define the anthropometric vertical axis (section 2.3.2). Each input image was pre-normalized with respect to *n-sn* landmarks. The facial expression recognition process fits the model starting with a "correct" initialization (model feature positions matched image feature positions). The images we performed the experiments exposed single AUs and additive AU combinations that built mainly prototypical expressions. The logic behind classification in images with AU combinations is similar with the one used in [Tia01]. For example, if we obtained the recognition results with AU12=0.7 and AU26=0.65 for a labeled AU12, it was treated as AU12+AU26. This means AU12 is recognized and AU26 is a false alarm.

Person-dependent experiments

We tested the 3D AMB AAM's ability to recognize facial expressions in subsets of images from FG-NET and MMI databases, showing facial expressions of persons present in the training dataset. The recognition results of the active model classifier are presented in Table 3.3.

AU	Signification	No.	Correct	False	Missed	Confused	Recognition Rate
	Neutral	20	19	1	0	0	95%
1	Inner Brow Raiser	24	21	0	3	0	87.5%
2	Outer Brow Raiser	52	41	1	9	1	78.8%
4	Brow Lowerer	42	41	0	0	1	97.6%
12	Lip Corner Puller	51	48	3	0	0	94.1%
15	Lip Corner Depressor	18	17	0	1	0	94.4%
26	Jaw Drop	74	55	0	19	0	74.3%
Total		281	242	5	32	2	86.1%
False Alarm: 1.7%, Missed: 11.3%							

Table 3.3: AU recognition for the person-dependent, static-based experiments.

The overall mean accuracy of the person-dependent system is 86.1%. This represents the recognition rate that was achieved when the training images contained faces of subjects present in test sequences. However, the experiment was person-dependent not person-and-expression-depended. That means that the persons can perform different expressions in the testing set compared to the training set. The main challenge is to see if the performance can be kept in person-independent experiments. Fig. 3.8 illustrates the synthesized facial expressions for known persons displaying combined AU expressions (from MMI database). The first two rows illustrate successful recognitions of combinations AU1+AU2+AU26 ("surprise" expression). The third row shows a miss of AU26 in an image labeled as AU12+AU26.

Fig.3.8: Synthesized expressions for known persons (2 successes, 1 missed)

Person-independent experiments

We tested the 3D AMB AAM's ability to recognize facial expressions of novel persons in images from MMI database showing single and AU combinations. The performance of the system is illustrated in Table 3.4.

AU	Signification	No.	Correct	False	Missed	Confused	Recognition Rate
	Neutral	20	17	3	0	0	85.0%
1	Inner Brow Raiser	10	8	0	2	0	80.0%
2	Outer Brow Raiser	27	20	1	5	1	74.0%
4	Brow Lowerer	24	21	1	1	1	87.5%
12	Lip Corner Puller	17	13	0	4	0	76.4%
15	Lip Corner Depressor	13	10	1	2	0	76.9%
26	Jaw Drop	24	14	0	10	0	58.3%
Total		135	103	6	24	2	76.2%
False Alarm: 4.4%, Missed: 17.7%							

Table 3.4: AU recognition for the person-independent, static-based experiments.

The overall mean accuracy of the static-person-independent system is 76.2%. Fig. 3.9 illustrates the synthesized facial expressions for known persons displaying combined AU expressions (from MMI database). The first two rows show successful recognitions of combinations AU1+AU2+AU26 ("happiness" and "fear" expressions). The third row shows a confusion of AU15 with AU1+AU26 ("fear" expression) in an image labeled as AU15 ("sadness" expression). The performance decrease can be explained by the fact that generally an AAM cannot synthesize expressions of faces with characteristics (shape and appearance) unseen in the training set (i.e. the method cannot synthesize "anger" of a person with facial hair and glasses if there are no similar faces in the training set).

Fig.3.9: Synthesized expressions for unknown persons (2 successes, 1 missed)

Chapter 4

Face Tracking Algorithm

A high-performance face tracking algorithm is one of the most important modules of a MBVC system. This chapter discusses a novel 3D head tracking method allowing for real-time recovery of rigid and non-rigid facial motion parameters in a monocular image sequence of a moving head. The described method uses a recursive motion estimation algorithm, namely an Extended Kalman Filter (EFK) to extract the head pose (global motion) and the developed 3D AMB AAM to extract the facial expressions (local motion). The resulting motion tracking system works in a realistic environment without makeup on the face, with an uncalibrated camera, and unknown lighting conditions and background.

4.1. STATE OF THE ART

A tracking system is a system that estimates the rigid and/or non-rigid motion of an object through an image sequence. The problem is technologically difficult, as 3D motion parameters have to be extracted from a 2D video sequence. The knowledge of the 3D scene and imaging process could be used to increase the accuracy and efficiency of the tracking. In the past several years, work has concentrated mainly on to recovering both 3D-head pose (rigid motion) [Aza93], [Lih93], [Jeb97], [Bas96], [Fuk93], [Cor02], [Str02], [Mor03] and facial expressions (non-rigid motion) [Lih94], [Bla95], [Ess95], [Nur98], [Ahl03], [Dor03] from a sequence of images of performer's face. The effects of head motion and facial expressions are combined in these images, so it is crucial to successfully separate the rigid from the non-rigid motion of the head (the problem known as "pose/expression separation"). The head pose has to be accurately computed before attempting to recover the expressions. There are two types of tracking systems [Ahl99]:

1. Motion-based
2. Model-based

Motion-based approaches are low-level tracking methods that rely on grouping movements of entities (points, lines, regions etc.) over time. They are fast but don't impose any semantic meaning to the tracked entities. Model-based approaches impose high-level semantic knowledge and low-level motion constraints. They add to the computation effort because of rigid and elastic model transformations (scaling, translation, rotation and deformation), but in the other hand increase the tracking accuracy and efficiency. In both approaches tracking is achieved using measurements given by geometric properties (points, lines, regions etc.) of the tracked object.

A tracking system is composed of two modules:

1. *Tracking module*, delivering the entity measurements. In case of point tracking the measurements are the 2D point positions $p_i(u_i, v_i)$, where $i=1,...,m$, and m is the number of measurement points.
2. *Estimator module*, delivering a state vector describing the 2D or 3D motion of the tracked object: $s = (translation, rotation, scale, deformation, ...)$

4.1.1. MOTION-BASED TRACKING

A motion-based tracker estimates the displacements of pixels in successive frames. The displacements are calculated using optical flow, block-matching techniques etc. The estimated motion field computes the model motion using optimization methods such as least square methods or extended Kalman filtering. The problem of motion-based trackers is the error accumulation that slowly leads to losing the track without the capability to recover. This is the so called "long sequence motion problem" [LiH93]. There are three main motion-based tracking techniques [Hor86], [Sha95].

1. *Feature-based* methods extract image features and track their movement from frame to frame to establish a global correspondence. The correspondence is used to estimate the geometric transformation between frames. Image features are low-level image descriptors, such as "regions," "edges," and "point features." Reliable tracking of *regions* is often difficult, since minor changes between frames can lead to very different segmentation in

consecutive frames. Arbitrarily curving *edges* are difficult to describe and track. Trackers based on *point features* such as nostrils, corners of eyes, mouth endpoints, and tips of eyebrows are increasingly used in computer vision applications [Hwa89], [Shi95]. However, in arbitrary scenes, the image noise and spatial and temporal sub-sampling can make motion and acceleration estimation difficult.

2. *Optical-flow* methods use spatial and temporal partial derivatives to estimate the image flow at each location in the image. Algorithms for recovering optical flow [Hor86] are based on a set of assumptions about the world that, by necessity, are simplifications and hence may be violated in practice, resulting in gross measurement errors. Moreover, the extraction of the optical flow from an image sequence is a highly computational task.
3. *Correlation-based* methods are popular for tracking objects [Hag96], [Jeb96]. They use the sum of the absolute differences between template and search area pixel intensities as a difference measure. On the negative side, the correlation tracking methods are sensitive to changes in overall illumination changes between frames of the sequence.

Morency et al. [Mor03] use 3D view-based eigenspaces, and a Kalman filter to calculate the head pose change in image sequences. They increase the tracking robustness by adaptive key frames, created from a single view of the person.

Azarbayejani et al. [Aza93] tracks the head position with feature points projected on an ellipsoidal model. Feature tracking often fails when the points are lost due to occlusions or lighting variations. Azarbayejani and Pentland [Aza95] tracking system recovers the 3D position and orientation of a human head the camera focal length, and the point depths by using an EKF. Continuing the work from [Aza95], Jebara and Pentland [Jeb97] use a feature-based tracking and an EFK to track the 3D head pose. The key facial features such as eyes and mouth corners are automatically detected in first frame. The 3D position of the features is estimated over the image sequence with an EKF and filtered using Eigenfaces to detect tracker failures.

4.1.2. MODEL-BASED TRACKING

A model-based tracker stores a model of the object to be tracked (appearance- and/or shape-based). Each frame the tracker changes the model parameters (position, shape, appearance etc.) in order to obtain the best fit. In this case the motion estimation is dependent on the object model and the new frame. This type of tracker can overcome the "long sequence motion" and recovery problems. The quality of the tracker is dictated by the flexibility of the model to cope with different appearances (pose, expression, illumination variations etc.) in new frames. Model-based trackers can also be:

1. Deterministic
2. Statistical

A deterministic tracker stores an accurate model of the tracked object hence it can generate precise results. There are several disadvantages when using this tracker:

1. *Appearance lack of flexibility*: the appearance model generates problems caused by changes in appearance due to variations in illumination, expressions etc.
2. *Significance of initialization*: The initialization parameters are critical. The tracker starts with the conditions (occlusion, illumination, expressions etc.) captured in the first frames, which may not be constant over the image sequence.

A statistical model-based tracker relies on a trained model to learn different aspects of appearance, shape etc. that is to be encountered during tracking. In the case of face tracking, the model could be trained on a database containing different faces exposing various facial expressions in different illuminations conditions. Lately, 2D AAM is increasingly used in statistical model-based trackers. Some 2D AAM implementations are able to track objects (faces, eyes, boxes etc.) using small-sized model fitting [Ste01], [Ahl03], [Han02]. Faces are tracked in a video stream at frame rates in range of 5 - 10 frames/second, with a low "shape-free" model resolution (64x64 pixels or less). Usually, AAM tracking applications use a multi-resolution (multi-scale) image or model. The tracking starts at low-resolution (coarse) levels and employs finer resolution images or active models at later stages of the search. However, if no temporal filtering is performed the robustness of the tracking is affected by pose, illumination, and speed changes.

Ahlberg [Ahl99], [Ahl03] tracks the non-rigid motion of a human head at the speed of 5 Hz, using 2D AAM and a weak-perspective camera model. The tracker does not employ any motion estimator, so that it can not follow rapid motions. Also, the tracker used only a frontal view AAM; hence it does not deal with head rotations.

The major problem of the classical appearance-based adaptation during tracking is the high computational time resulting from the inclusion of a synthesis step in the iterative optimization. Good performance needs a large face subspace, which in return decreases the algorithm speed. Thus a real-time performance cannot be achieved.

Sclaroff et al. [Sca98], [Cas98] developed "Active Blobs" for tracking, which is a *deterministic model-based* tracker. Sclaroff uses a 3D non-rigid, textured cylinder as a head model, which is initialized in the first frame of the tracking sequence. Similarly to the AAM fitting, image differences drive the tracking by off-line learning the relationship between image errors and parameter offsets. However, the Active Blobs are derived from a single example, whereas AAM uses a training set of examples to model shape and intensity variations. The tracking exposes the common problem of the deterministic tracker that is failure due to error accumulation.

DeCarlo and Metaxas [DeC96] use a manually positioned wireframe head model to extract facial shape and expressions. Their method calculates optical flow at feature points and regularizes it using the model motion. They employed a Kalman filter and edge information to stabilize the measurements affected by optical flow error accumulation.

Basu, Essa and Pentland [Bas96] use a fusion of the optical flow and a 3D ellipsoidal model to track faces. Since the method doesn't use known facial points to anchor the 3D model to the face the flow errors accumulate and drifts off the face tracker.

Strom's [Str99], [Str01], [Str02] SFM tracking system continues the work of Azarbayejani et al. [Aza95], and Jebara et al. [Jeb97]. Strom uses a 3D face model and an ABS module to achieve robust performance in tracking the 3D pose of a human head.

4.2. A COMBINED TRACKING SYSTEM

The objective of this section is to build a real-time system for 3D face tracking and modeling. The first step in a full automatic model-based tracking system is the face detection allowing the identification and location of the face in first image frames. The next step is motion estimation encompassing global and local motion recovery, expression and emotion analysis, etc. The problem is not trivial, as 3D motion parameters have to be extracted from a sequence of 2D images of the performer's head-and-shoulders. The human face is a complex dynamic system, and the facial features are time-varying and noisy signals. A tracking system trying to recover rigid or/and non-rigid facial movement requires the use of a recursive estimation framework.

The method we propose builds on SFM estimation framework created by Azarbayejani to estimate 3D motion and structure of objects from images [Aza93], [Jeb99], [Str99]. A similar system recovering the 3D human head motion was successfully developed and tested earlier in [Cor02]. The new developed system employs an EKF to fuse *feature-motion- and statistical-model-based* tracking approaches, using a full projective camera model. The system recovers the rigid and non-rigid facial motion parameters in monocular image sequences. The main purpose of the EKF is to filter and predict the rigid and non-rigid facial motion. The feedback from the EKF imposes a global collaboration among the separate 2D feature trackers. The presented solution enables fast and stable on-line tracking of extended sequences, despite noise and large variations in pose.

Adding a 3D face model to a motion-based tracker has several advantages [Jeb02]:

1. *Self-Occlusion Prediction*: the 3D model is used to predict feature occlusions, this way determining the reliability of feature measurements.
2. *Feature Position Control*: the features to be tracked are obtained by rendering the model in the estimated pose. This way rendered 3D mode vertices help obtaining feature points and infer their occlusion during tracking.
3. *Texture Update*: a full 3D textured head model can be recovered.
4. *3D Structure Constraint*: the 3D model fuses information of 3D structure and 2D measurements at each frame. The positions of the feature points affected by noise are constrained using the 3D model.

Our model-based tracker employs the 3D AMB AAM [Cor06] to exploit all advantages mentioned above. Additionally, the inclusion of the 3D AMB AAM in the estimation procedures adds a set of constraints that are crucial for stable and real time motion estimation. The 3D AMB AAM encodes anatomical knowledge about facial expressions, and facial structure. The 3D facial shape is recovered first by molding the active face model onto the face image using deformations constrained by anthropometrical statistics. Once the generic active model is transformed into a personalized one, the anthropometric parameters may be kept constant, allowing only for facial expression deformations. To fit a generic face model to a specific face, both the global and local motions have to be recovered. In our approach we decouple the global head movements and local non-rigid deformations. This separation is achieved in two stages:

1. Estimation of the global (rigid) motion using a robust statistical estimator (EKF)
2. Local adaptation of the 3D model to fit possible facial animations (3D AMB AAM).

Varied facial appearances and shapes make it particularly difficult to estimate the local transforms. Once the global motion is recovered we estimate the facial expressions in a frontal, shape-free synthesized face view using the 3D ABM AAM and the described AAM search technique (Section 3.1). The AAM searches the space of a trained pool of facial expressions, to locally adapt the 3D model. The main task of the Kalman filter in this implementation is to filter and predict global and local motion, hence to ease the load of the AAM search error. In other words, by predicting the pose and facial expression in the next frame, EKF reduces the active model search space.

The proposed tracking system is capable of coping with large variation of pose even though we trained the active model on frontal or near-frontal views. This is possible because of shape representation in 3D, which allows model de-rotation and projection to 2D for any given pose. In all the experiments, the model has demonstrated a reliable performance between [-45, 45 degrees] in yaw angle. However, for poses near or over 60 degrees the tracking becomes less reliable due to the missing information of the rotated view.

4.3. EKF FOR GLOBAL MOTION ESTIMATION

The general problem of recovering 3D position parameters from 2D images could be solved using different 2D views of the 3D objects. If these images are taken at the same time, the problem is solved by *stereovision* [Fau92] or *trifocal tensor* [Har94]. Another approach using monocular 2D images of moving objects is known as *structure-from-motion* (SFM) [Aza95]. The problem can be formulated as a parameter estimation problem: "Given a number of noisy measurements of 2D-tracker positions, we have to optimally recover the SFM components of equation 4.1". Least-squares techniques have been popular to solve such type of problems, in computer vision. Unfortunately, these techniques need a priori knowledge of the 3D object models and are not able to handle gross and systematic errors, and correlation in measurements. These unrealistic assumptions prevent least-squares methods from converging to an acceptable solution. A robust alternative to the least square methods is the Extended Kalman Filter (EKF), for the following reasons [Zha92]:

1. EKF explicitly considers the observation uncertainties.
2. EKF is fed with measurement recursively.
3. EKF is a simple and robust solution to parameter estimation problems.

Generally, a Kalman filter is a versatile recursive state estimator, which is a tool that refines an initial estimation of the state while new observations are acquired. The literature on Kalman Filter-based recursive algorithms for recovering 3D structure and/or motion from a monocular sequence of images dates back to the early eighties [Gen82]. Extensive work has been done on three main areas of the problem:

1. recovery of motion with known structure [Aza93], [Bro86], [Gen82], [Gen92]
2. recovery of structure with known motion [Mat89], [She92], and
3. recovery of motion and structure simultaneously [Aza95], [Jeb96], [Bro91].

Gennery [Gen82], [Gen92] considers the problem of tracking 3D objects of known shape, and recovering object motion and position with respect to the observer. His state vector therefore consists of 12 parameters (6 for pose and 6 for velocity representation). For tracking he uses both point and line features.

Matthies et al. [Mat89] make the assumption of known motion, and demonstrate their scheme in the special case of translational motion both with a feature-based scene representation and with a dense depth-map. Shekhar and Chellappa have successfully tested a similar approach, again assuming known motion [She92]. Matsumoto and Zelinsky [Mat99] simplified the SFM problem formulation using an EKF in a stereo-based face tracking.

Given 2D-object images the SFM problem aims to recover:

1. the 3D object coordinates
2. the relative 3D camera-object motion
3. camera geometry (camera calibration)

EKF is used to solve the SFM problem resulting in an accurate, stable and real time solution. The EKF takes in consideration the non-linear aspect of perspective mapping between the 3D world and its projection.

The face tracking problem should be formulated as dynamic parameter estimation of a stochastic process where identity and appearance parameters are kept constant and expression parameters change freely. Temporal frameworks such as Kalman filters provide a recursive solution to this problem. In the next section we present an EFK based technique, used to recover 3D motion parameters, camera focal length, and facial expressions. The EKF provides a temporal estimation of model parameters and including those related to facial expressions provide a more robust and stable fit over time.

The developed EKF-SFM framework consists of two main modules:

1. *Tracking module*, delivering the 2D point measurements $p_i(u_i, v_i)$ of the tracked features, where $i = 1,...,m$ and m is the number of measurement points.
2. *Estimator module* (for the estimation of 3D motion and expressions), delivering a state vector

$$s = (t_x, t_y, t_z, \alpha, \beta, \lambda, f, a_j) \tag{4.1}$$

where $(t_x, t_y, t_z, \alpha, \beta, \lambda)$ are the six 3D camera/object relative motion, namely translation and rotation, f is the camera focal length, and a_j are the 3D model's actuators (muscles) that generate facial expressions, where $j = 1,...,n$ and n is the number of actuators. The set of 2D features to be tracked is obtained by projecting their corresponding 3D model points $P_i(X_i, Y_i, Z_i)$, where $i = 1,...,m$, and m is the number of tracked features. Anthropometric actuators can be included in the state vector if structure estimation is needed.

In this formulation the observation motion vector is expressed from the 2D projection of the 3D head features. The facial expression parameters are fully observable, hence their corresponding part of observation and state vector are identical.

Tracking should allow estimation of the motion while locating the face. Before tracking, a set of facial features to be tracked has to be detected or selected. Since we are trying to recover rigid head motion we choose facial features affected only by the rigid motion. In other words, since we separate the global from local motion search spaces, the features to be tracked must originate from rigid regions of the face. The (u,v) measurements depend on actuator intensities only if the projected 3D coordinates are in the influence zones of the muscles. We choose rigid facial points for pose tracking, so that there is no correlation between (u,v) features and muscles a_j. Using the 3D model we have the control to add only rigid points located in "muscle-free" zones, such as: eye sockets, nostrils, ears etc. The 3D pose estimate is used to constrain the search area for the 2D feature matching. The structure estimation (given by actuator-caused deformations) helps to transform the generic model into an individual one, namely by molding the 3D model to better match the individual head structure. The EKF provides a convenient mechanism for fusing all the information about the reliability of the measurements into an estimate of the 3D structure, the pose and the focal length of the camera. Similarly to [Cor02] we employ a *feature-motion-based* tracking technique to obtain the 2D observations, which EKF can use to infer the 3D information.

The EKF, detailed in Appendix A, is applied to nonlinear systems and consists of two stages: time updates (or prediction) and measurement updates (or correction). At each iteration, the filter provides an optimal estimate of the current state using the current input measurement, and

produces an estimate of the future state using the underlying state model. The values, which we want to smooth and predict independently, are the tracker state parameters. The EKF state and measurement equations and can be expressed as:

$$s(k+1) = As(k) + \xi(k) \tag{4.2}$$

$$m(k) = Hs(k) + \eta(k) \tag{4.3}$$

where s is the state vector, m is the measurement vector, A is the state transition matrix, H is the Jacobian that relates state to measurement, and $\xi(k)$ and $\eta(k)$ are error terms modeled as Gaussian white noise. The observations are the 2D feature coordinates (u,v) and a_i which are concatenated into a measurement vector $m(k)$ at each time step. The observation vector is the back-projection of the s state vector containing the relative 3D camera-scene motion, and the camera focal length. In our case the state vector is $s(translation, rotation, velocity, focal\_length, actuators)$ that contains the relative 3D camera-object translation, rotation and their velocities, camera focal length, and the actuators (muscles) that generate the facial expressions.

In this projective camera framework s to m mapping is nonlinear ($H(s)$ varies with s), and corrupted by some noise encoded in the time-varying covariance matrix $R(k)$. Large variances $R(k)$ represent lost 2D features during the tracking process. The system evolution matrix A is based on first order Newtonian dynamics in 3D and assumed time invariant. The 3D structure, motion and the focal length are constrained through the A matrix, which sets a linear dependency between the past and current values. The internal state contains a random Gaussian noise $\xi(k)$, representing the variations of the internal state.

The EKF requires a physical dynamic model of the motion and a measurement model relating image feature locations to motion parameters. Additionally, because this approach recovers the motion without structure, a representation of the object (user's head) is required. In this section we present the chosen motion and measurement models and their resultant equations.

4.3.1. THE MOTION MODEL

The dynamic model of the head tracking is a discrete-time Newtonian physical model of a rigid body motion, moving with constant velocity. The state vector:

$$s(t_x,t_y,t_z,\omega_x,\omega_y,\omega_z,f,\dot{t}_x,\dot{t}_y,\dot{t}_z,\dot{\omega}_x,\dot{\omega}_y,\dot{\omega}_z,a_j)$$

consists of 13 elements related to pose and focal length, grouped as follows: the relative camera-object translation (t_x,t_y,t_z), the small inter-frame rotation $(\omega_x,\omega_y,\omega_z)$, the camera focal length f, the translational velocity $(\dot{t}_x,\dot{t}_y,\dot{t}_z)$, and the rotational velocity $(\dot{\omega}_x,\dot{\omega}_y,\dot{\omega}_z)$. Additionally the state vector includes the facial expression actuators that we wish to "smooth": $(a_{1-4}$ = Jaw Drop, Mouth Corners, Eyebrow Middle, Eyebrow Inner).

The state equation 4.2 could be written as:

$$\begin{pmatrix} t_i \\ \omega_i \\ f \\ \dot{t}_i \\ \dot{\omega}_i \\ a_j \end{pmatrix}_{k+1} = \begin{pmatrix} I & 0 & 0 & I\Delta\tau & 0 & 0 \\ 0 & I & 0 & 0 & I\Delta\tau & 0 \\ 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{pmatrix} \cdot \begin{pmatrix} t_i \\ \omega_i \\ f \\ \dot{t}_i \\ \dot{\omega}_i \\ a_j \end{pmatrix}_k + \xi(k) \qquad (4.4)$$

where $i = x,y,z$ is the index of the coordinate axes of the camera reference frame, I is the identity matrix and $\Delta\tau$ is the inter-frame time. This work employs a simplified state vector, without considering the translational and rotational velocities.

4.3.2. THE MEASUREMENT MODEL

The measurement model relates the state vector s to the 2D-image location (u_k,v_k) of each image feature point p_k. The point $p_k(X_k,Y_k,Z_k)$ of the object reference frame becomes the point $p_{ck}(X_{ck},Y_{ck},Z_{ck})$ of the camera reference frame, where:

$$\begin{pmatrix} X_{ck} \\ Y_{ck} \\ Z_{ck} \end{pmatrix} = T(t_x, t_y, t_z) + R(\alpha, \beta, \gamma) \begin{pmatrix} X_k \\ Y_k \\ Z_k \end{pmatrix}, \; k = 1,..,N \tag{4.5}$$

where T and R represent the object (or camera) translation and rotation matrices, (α, β, γ) are the Euler angles, and N is the number of points. The observed perspective projection is given by:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \frac{f}{Z_{ck}} \begin{pmatrix} X_{ck} \\ Y_{ck} \end{pmatrix} \quad , \; k = 1,..,N \tag{4.6}$$

where f is the camera focal length.

The above equation is valid when considering the camera system origin in center of projection (*COP*). If the system origin is fixed at the image plane, the above equation transforms to:

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \frac{1}{1 + \beta Z_{ck}} \begin{pmatrix} X_{ck} \\ Y_{ck} \end{pmatrix} \quad , k=1,...,N, \tag{4.7}$$

where $\beta = \frac{1}{f}$ is the inverse focal length. The equivalence can be easily verified, by replacing Z_{ck} in (4.6) with $f + Z_{ck}$. The choice of the projection equation will reflect in the calculation difficulty of the H Jacobian, at each frame step.

At each filter cycle we have to calculate the partial derivatives of measurements with respect to each of the unknown parameters. [Low87] proposed a re-parameterization of the projection equations, to simplify the calculation of H Jacobian, by expressing the translations in the camera coordinate system rather than model coordinates. In this case the measurement equation will take the following form:

$$\begin{pmatrix} X_{ck} \\ Y_{ck} \\ Z_{ck} \end{pmatrix} = R(\alpha,\beta,\gamma)\begin{pmatrix} X_k \\ Y_k \\ Z_k \end{pmatrix} \tag{4.8}$$

$$\begin{pmatrix} u_k \\ v_k \end{pmatrix} = \begin{pmatrix} \frac{f}{Z_{ck}+t_z}X_{ck}+t_x \\ \frac{f}{Z_{ck}+t_z}Y_{ck}+t_y \end{pmatrix} \tag{4.9}$$

The first partial derivative (Jacobian) of each measurement relative to each state variable used in the EKF minimization process has the form:

$$H_{i,j} = \left(\frac{\partial h_i(s)}{\partial s_j}\right), i = 1,...,2N\ ,\ j = 1,...,M \tag{4.10}$$

where s is the M-component state vector, and k is the index of the u or v feature coordinates in the observation vector $m_k(u_1,v_1,u_2,v_2,...,u_N,v_N)$. We start the calculation of the Jacobian H from the above nonlinear, but differentiable and well-behaved equations (4.8) and (4.9).
In the Lowe's model, the object's coordinates in the camera reference frame X_{ck}, Y_{ck}, and Z_{ck} are defined as:

$$(X_{ck},Y_{ck},Z_{ck}) = (r\cos\omega_z, r\sin\omega_z, z) \tag{4.11}$$

where r is the distance of the point $p_{ck}(X_{ck},Y_{ck},Z_{ck})$ from z axis. An advantage of including the small inter-frame rotations $(\omega_x,\omega_y,\omega_z)$ in the state vector is that the derivatives of points in camera frame $p_{ck}(X_{ck},Y_{ck},Z_{ck})$ with respect to the small rotations can be calculated very simple from (4.10). For instance the derivative of camera coordinate X_{ck} with respect to the rotation around z axis ω_z is $\frac{\partial X_{ck}}{\partial \omega_z} = -r\sin\omega_z = -Y_{ck}$. This will result in a simple form of the Jacobian sub-

matrix related to rotations. Table 4.1 gives a summary of the partial derivatives of the X_{ck}, Y_{ck} and Z_{ck} with respect to the small rotation angles ω_x, ω_y and ω_z.

	∂X_{ck}	∂Y_{ck}	∂Z_k
$\partial\omega_x$	0	$-Z_{ck}$	Y_{ck}
$\partial\omega_y$	Z_{ck}	0	$-X_{ck}$
$\partial\omega_z$	$-Y_{ck}$	X_{ck}	0

Table 4.1: Partial derivatives of coordinates in camera frame with respect to the small inter-frame rotations.

Usually, it is difficult to calculate the partial derivatives for the standard form of the projection equation (4.6), (4.7) [Aza95], [Jeb99]. Lowe's [Low87] reparametrization (4.8), (4.9) greatly simplifies the Jacobian calculation. Considering p a state vector parameter, and using the derivation chain rule, we have:

$$\frac{\partial u}{\partial p} = \frac{f}{Z_{ck}+t_z}\frac{\partial X_{ck}}{\partial p} - \frac{fX_{ck}}{(Z_{ck}+t_z)^2}\frac{\partial Z_{ck}}{\partial p} \tag{4.12}$$

For simplicity, we substitute:

$$d = \frac{1}{Z_{ck}+t_z} \tag{4.13}$$

Hence:

$$\frac{\partial u}{\partial p} = fd\left(\frac{\partial X_{ck}}{\partial p} - dX_{ck}\frac{\partial Z_{ck}}{\partial p}\right) \tag{4.14}$$

and similarly,

$$\frac{\partial v}{\partial p} = fd\left(\frac{\partial Y_{ck}}{\partial p} - dY_{ck}\frac{\partial Z_{ck}}{\partial p}\right) \tag{4.15}$$

If we substitute p in (4.8) and (4.9) with the state vector components, and use the Table 4.1 information, we get the Jacobian components as shown in Table 4.2.

Horizontal measurements u_k**:**	***Vertical measurements*** v_k**:**
$\frac{\partial u}{\partial t_x} = 1$	$\frac{\partial v}{\partial t_x} = 0$
$\frac{\partial u}{\partial t_y} = 0$	$\frac{\partial v}{\partial t_y} = 1$
$\frac{\partial u}{\partial t_z} = -fd^2 X_{ck}$	$\frac{\partial v}{\partial t_z} = -fd^2 Y_{ck}$
$\frac{\partial u}{\partial \omega_x} = -fd^2 X_{ck} Y_{ck}$	$\frac{\partial v}{\partial \omega_x} = -fd(Z_{ck} + dY_{ck}^2)$
$\frac{\partial u}{\partial \omega_y} = fd(Z_{ck} + dX_{ck}^2)$	$\frac{\partial v}{\partial \omega_y} = fd^2 X_{ck} Y_{ck}$
$\frac{\partial u}{\partial \omega_z} = -fdY_{ck}$	$\frac{\partial v}{\partial \omega_z} = fdX_{ck}$
$\frac{\partial u}{\partial f} = dX_{ck}$	$\frac{\partial v}{\partial f} = dY_{ck}$

Table 4.2: Jacobian point components with respect to the pose and focal length

When N points are tracked, there are $2N$ measurements (coordinates of point projections) at each frame and 7 parameters to be recovered (six motion parameters plus camera focal length). Both motion and focal length are over-determined at each frame when $2N > 7$, which happens when $N \geq 4$, i.e. when tracking 4 or more points. When camera parameters are known beforehand, we need $N \geq 3$ points to recover the 3D motion. Since state actuator parameters $(a_j, j = 1-4)$ are fully observable the Jacobian part related to these are easy to compute:

$$\frac{\partial a_j}{\partial a_i} = \begin{cases} 1, & if\ i = j \\ 0, & if\ i \neq j \end{cases}$$
$$\frac{\partial a_j}{\partial p} = 0, \tag{4.16}$$

where p is the part of the state vector related to pose or focal length parameters. The novel aspect of this approach is that the filter directly couples pixel measurements with the muscle actuators responsible for creating facial expressions.

We compute a three-parameter incremental rotation $(\omega_x, \omega_y, \omega_z)$, similar to that used in [Jeb96] to estimate inter-frame rotation. The $\omega_x, \omega_y, \omega_z$ parameters are known as incremental Euler angles. Unbiasing (i.e. centering about the zero mean) these incremental Euler angles, avoids over-parameterization. Being approximately independent, they can be used reliably in system linearization [Jeb99]. An incremental rotation quaternion can be computed from these three parameters:

$$\delta q = \left(\sqrt{1-\eta}, \frac{\omega_x}{2}, \frac{\omega_y}{2}, \frac{\omega_z}{2} \right) \tag{4.17}$$

$$\eta = \frac{\omega_x^2 + \omega_y^2 + \omega_z^2}{4} \tag{4.18}$$

The incremental rotation computed at each frame step is combined into a global quaternion vector (q_0, q_1, q_2, q_3) used in the *EKF* linearization process and rotation of the 3D model [Sho94]. A *3x3*-rotation matrix *R* can be generated using unit quaternions:

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{4.19}$$

Figure 4.1 illustrates the integration of this small-rotation algorithm in the recovery of the global rotation of the 3D head model.

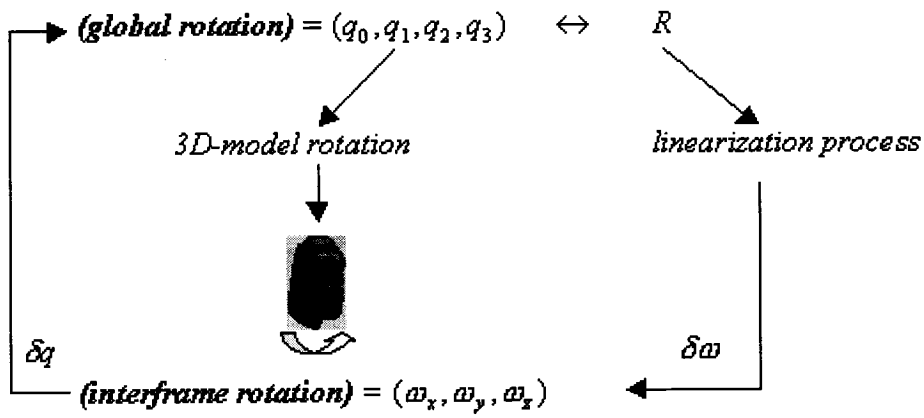


Fig. 4.1: The rotational process in 3D-pose recovery algorithm.

4.4. A 3D AMB AAM FOR LOCAL MOTION ESTIMATION

The output of the global motion estimation is used as a starting point for local motion detection. The optimal EAU values are determined based on the 3D AMB AAM ABS fitting procedure (Section 2.3.5). The basic idea is to analyze the images from the differences between a real image and a corresponding synthetic image generated by the 3D AMB AAM to estimate the transformation parameters (Fig. 4.2).

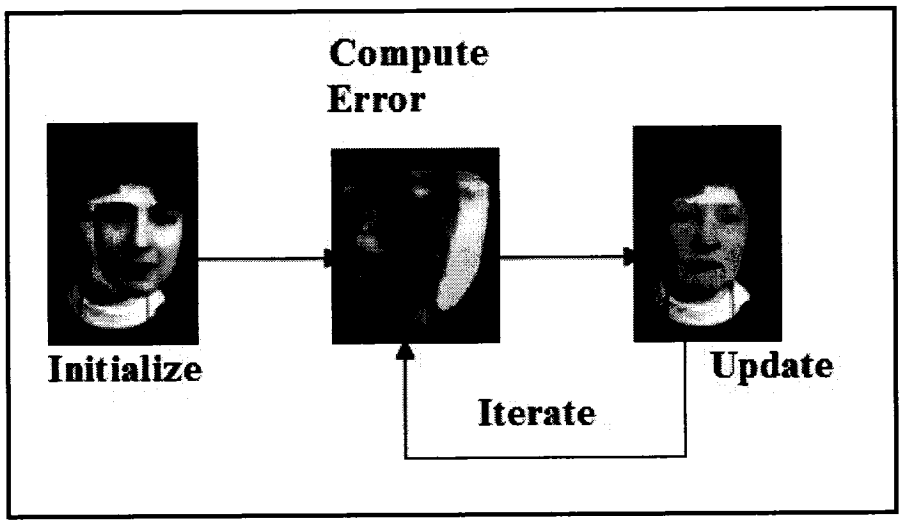


Fig. 4.2: Fitting faces with an AAM (from [Coo98])

In each frame the active model detects the following EAUs (Section 2.3.1):

1. a_1: Jaw Drop
2. a_2: Mouth Corners
3. a_3: Eyebrow Middle
4. a_4: Eyebrow Inner

The model fitting starts with the pose and expression values estimated by the Kalman filter. Thus for a given sequence, the vector $(t_x, t_y, t_z, \alpha, \beta, \lambda, a_1, a_2, a_3, a_4)$ describes the time dependent pose and animation of the 3D wireframe model. These values are used to initialize the 3D AMB AAM fitting procedure. The fitting algorithm will output updated values for facial pose and expression. In the tracking context, the fitting results associated with the current frame will be handed over to EKF to compute the estimates for the next frame (Fig. 4.3).

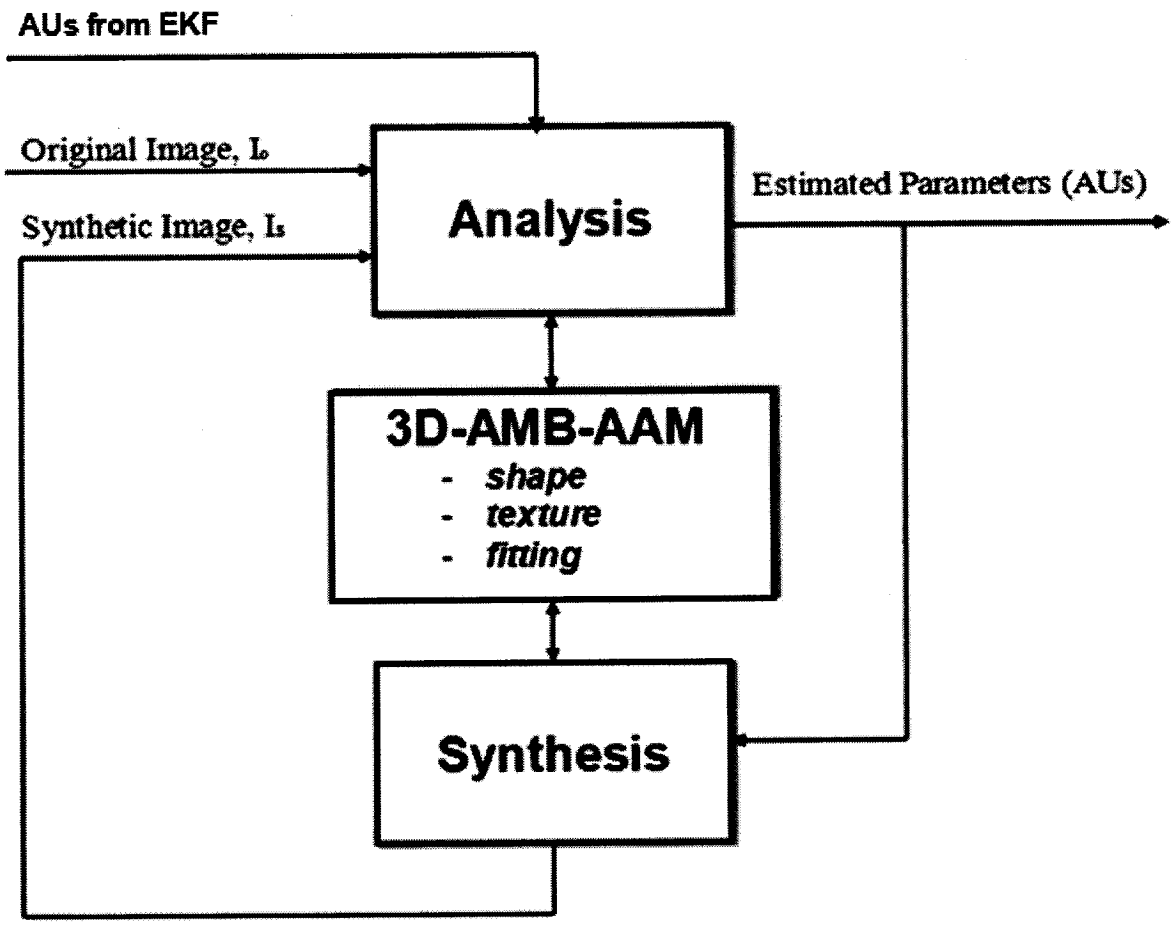


Fig. 4.3: Analysis-by-Synthesis module of the Extended Kalman Filter.

As shown in Section 3.1 (*Facial Image Segmentation*) the active model search starts with a known pose and a known *AE shape* vector composed of facial types and expressions. The fitting process needs the facial pose and *c* parameters that define the combined AE shape and texture space (equation 2.18). At each step a correction ∂d is found based on image difference. The correction represents small displacements in *pose* and *c* parameters for best image fitting. The output represents an updated pose and *c* parameter vector. The updated facial expressions a_j can be obtained from the *c* parameters using 2.19. The updated a_j values are passed to the EKF for filtering and estimation. The estimated expression a_j^+ parameters from the EKF are transformed back to the *c* parameters using equation 2.20. The new *c* parameters are used for a new ABS fit, and the cycle continues. The ABS updating process is illustrated in detail in Fig. 4.4. *Texture Instance ( )* and *Shape Instance ( )* are the main function components of the synthesis phase. *Sample Shape ( )* is the main function component of the analysis phase. The *Texture Instance ( )*

function takes as input the *c* parameters, and provides a normalized reconstructed texture using equation 2.19. The *Shape Instance ( )* function takes the same *c* parameters as input and reconstructs the facial type and expression of the tracked face, based on same equation 2.19. *Sample Shape ( )* takes as input the pose parameters and the reconstructed shape that is the output of *Shape Instance ( )*, deforms the 3D model accordingly, and then extracts the real texture by projecting the 3D model in the image plane. The output of *Sample Shape ( )* is the shape-free, real texture of the 3D model.

Without the use of EKF the fitted shape vector of the current frame is propagated to the next frame and used as an input for next image fit. In this case, because of large pose variations, fast motion, or simply local minima, the tracking will quickly fail. Integrating the ABS fitting procedure in the EKF recursive schema can result in robust expression tracking outcome. In conclusion, at each frame the expression part of the *AE shape* vector is replaced by the estimated EKF expression vector. Adding the EKF in the ABS estimation process is shown in Chapter 5 to greatly improve the expression tracking stability.

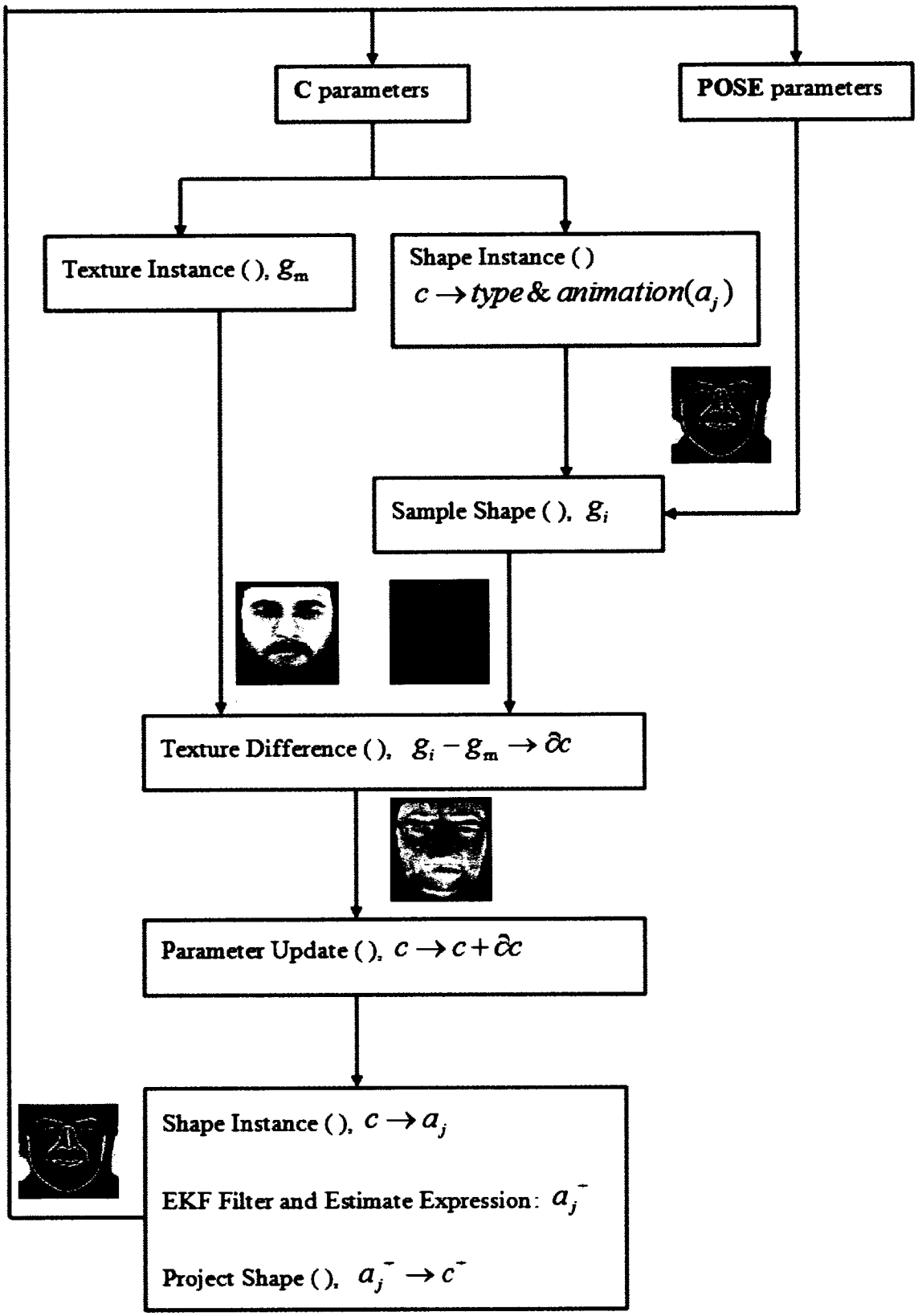


Fig. 4.4: Functional details of the Analysis-by-Synthesis module of the Extended Kalman Filter

The model fitting is performed in a geometrically normalized image, which represents a frontal view of the face. The geometry of the normalized image is obtained by projecting the generic wireframe mesh using a 3D frontal view onto an image with a specified resolution. The normalized image is basically a 2D capture of the frontal 3D textured model. During tracking we use the facial texture symmetry in the warping procedure, for yaw angles greater than 20 degrees. The texture-half to be mirrored is dictated by the yaw angle of the 3D model measuring the vertical face orientation. This way, the synthesized shape-free texture will be more accurate since severe distortions and occlusions are introduced by rotation in depth (Fig. 4.5).

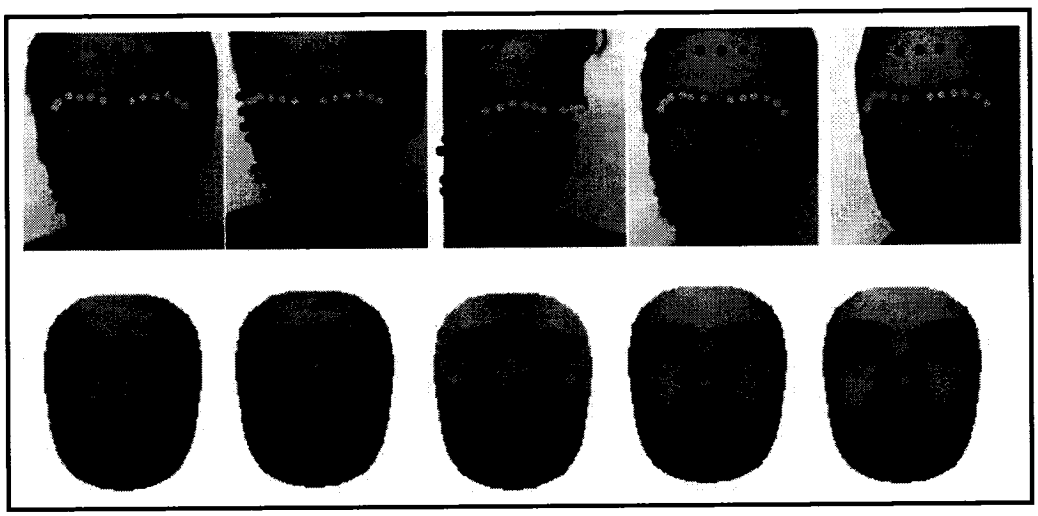


Fig. 4.5: Extracting the shape-free texture during tracking.

The next chapter presents experiments of our tracking algorithm in real and synthetic image sequences using ground-truth data. For this purpose we developed a fully automatic solution to the problem of 3D rigid and non-rigid motion tracking, called Automatic Face Tracking System (AFTS). The system integrates the newly proposed 3D AMB AAM and the novel 3D face pose and expression tracking algorithm.

Chapter 5

Face Tracking Validation

This chapter presents the test procedures of our tracking algorithm in real and synthetic image sequences using ground-truth data. Also, we developed an Automatic Face Tracking System (AFTS) that integrates the newly proposed 3D AMB AAM with the novel 3D face pose and expression tracking algorithm. Fitting the active model onto an image sequence frame by frame leads to a fluctuant estimation of the model parameters due to variation of pose, identity, expression, image noise, illumination, and local optima. The face tracking problem should assume the presence of a subject whose identity remains unchanged throughout the sequence. We implemented a fully automatic solution to the problem of 3D rigid and non-rigid motion tracking, described in the next sections. We used ground-truth marked sequences to objectively evaluate the tracking performance.

5.1. EKF INITIALIZATION

The implemented EKF is initialized automatically in three stages involving: Face Detection, Model Alignment, Point Selection, and Model Adaptation.

1. **Face Detection**: the face with arbitrary orientation is detected using a free implementation [Ope05] of Viola et al. [Vio01] face detection algorithm (Fig. 5.1 first row).
2. **Model Alignment**: using same method [Vio01], 3 facial points are detected (eyes and nose) in first frame to compute the initial face pose (Fig. 5.1 second row). These points are used as anchors in the initialization stage. After establishing 2D-3D point correspondence we coarsely align the 3D wireframe model to a 2D face image using the Weak-Perspective-3-Points (WP3P) technique [Alt92].

3. **Point Selection**: the 3D model provides the initial structure (X_i, Y_i, Z_i). Each 2D-feature point (u_i, v_i) corresponds to a structure point $p_i(X_i, Y_i, Z_i)$. The (u_0, v_0) initial points are obtained by intersecting the 2D image plane with a ray rooted in the camera's center of projection COP and aiming to the 3D structure point on the head model. Using the 2D-3D point correspondence the system selects 4 or more feature points for pose tracking. The points are chosen from the rigid region of the face, such as: eye corners, nostrils, ears etc. (Fig. 5.1 second row).
4. **Model Adaptation**: once the 3D model is aligned with the face to be tracked the first ABS fit is performed. The initial fit of the 3D AMB AAM finds the geometry of the face to be tracked (facial type), and the initial facial expression. The fitting process also samples the texture of the real face, which is mapped to the 3D face, for a realistic appearance (Fig. 5.1). This step simulates the initialization stage of the MBVC when the 3D model geometry and captured facial texture are sent to the receiver.

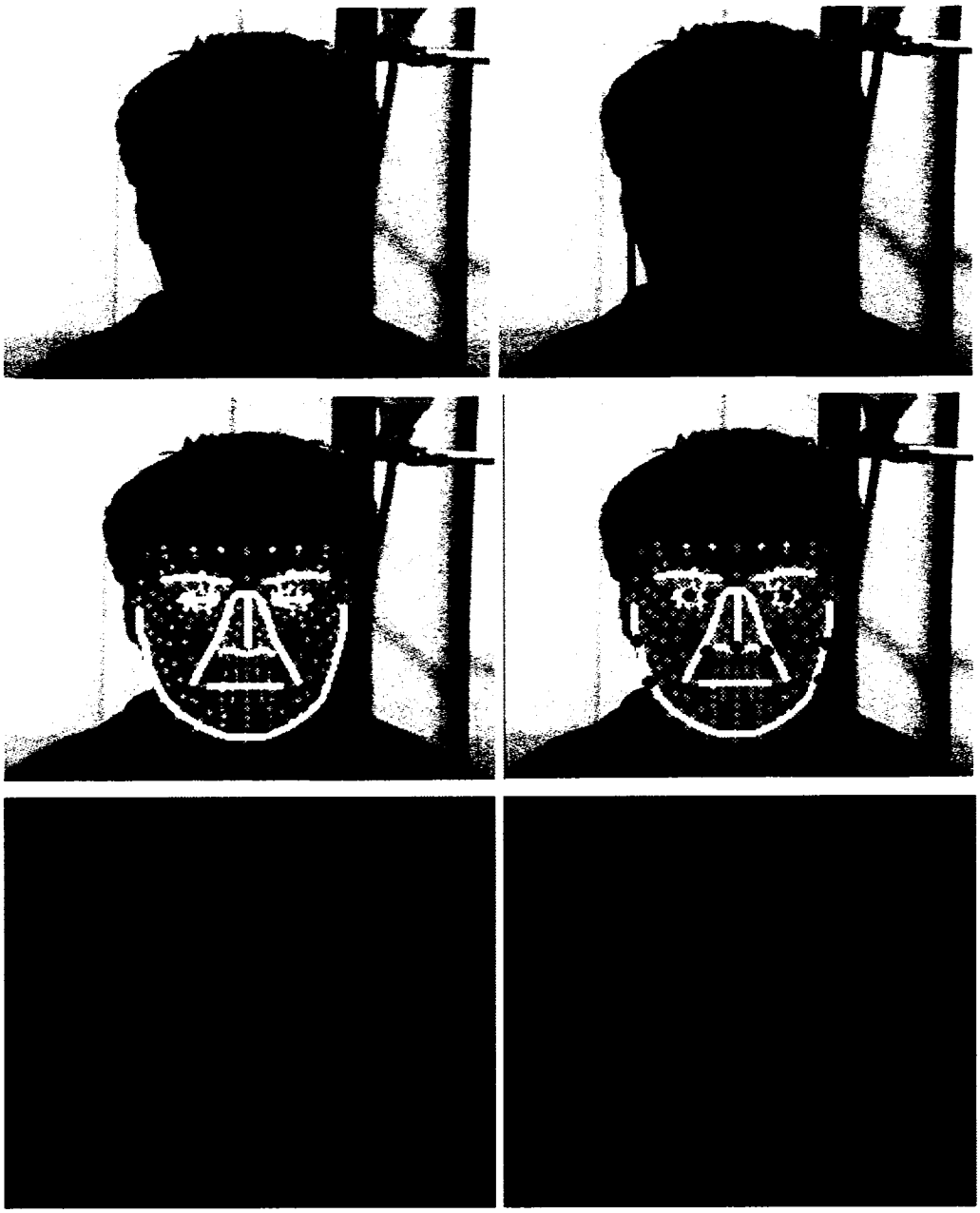


Fig. 5.1: Automatic initialization of the tracker (detection of anchor points in *first row*, 3D-2D model alignment in *second row*, extracted initial texture in *third row*). The face is from MMI database [Pan05]).

5.2. EKF UPDATE

The EKF update stage is illustrated in detail in Fig. 5.2. At each iteration, the EKF computes an estimate of the rigid 3D head motion. Also the EKF provides an estimate of the animation

parameters. We employ the Kanade-Lucas-Tomasi (KLT) [Shi94] 2D-gradient feature tracking method, which robustly performs the 2D tracking reinforced by the EFK estimation output. An estimate of rigid motion, camera focal length, and actuator values are found at each step. After state parameter recovery, the 3D pose and expression values are used as a starting point for the 3D AMB AAM, which performs a better face fit by adjusting model parameters (muscle actuators) and refining pose. Using the final 3D model fit, a perspective transformation will project feature points back onto the image to determine a corrected position of the 2D feature trackers. At the next frame in the sequence a 2D tracking is performed starting at this 2D estimated position. The current matching coordinates of tracked features are fed back into the Kalman filter as the observation vector, and the loop continues (Fig. 5.2, Fig. 5.3). The feedback from EFK is used to update the 3D model pose and animation parameters, i.e. provides the 3D head tracking information. Including a 3D model into this recursive framework has several advantages:

1. It compensates for 3D transforms (rotation, translation and scale), allowing the use of fast 2D feature matching to track the salient points
2. Decides the visibility of a feature point during tracking. When a feature point becomes occluded in the rendered image, its measurement can be discarded from the Kalman filter computation. Setting the variance for that feature to a high value in that step will lower its weight in the tracking process.

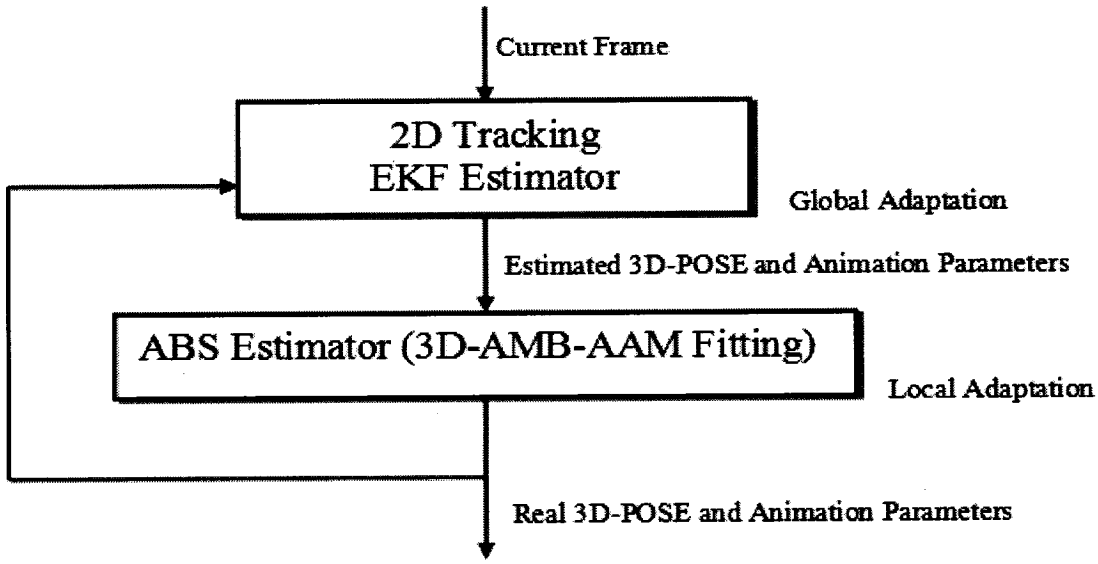


Fig. 5.2: Block diagram of the combined tracking system

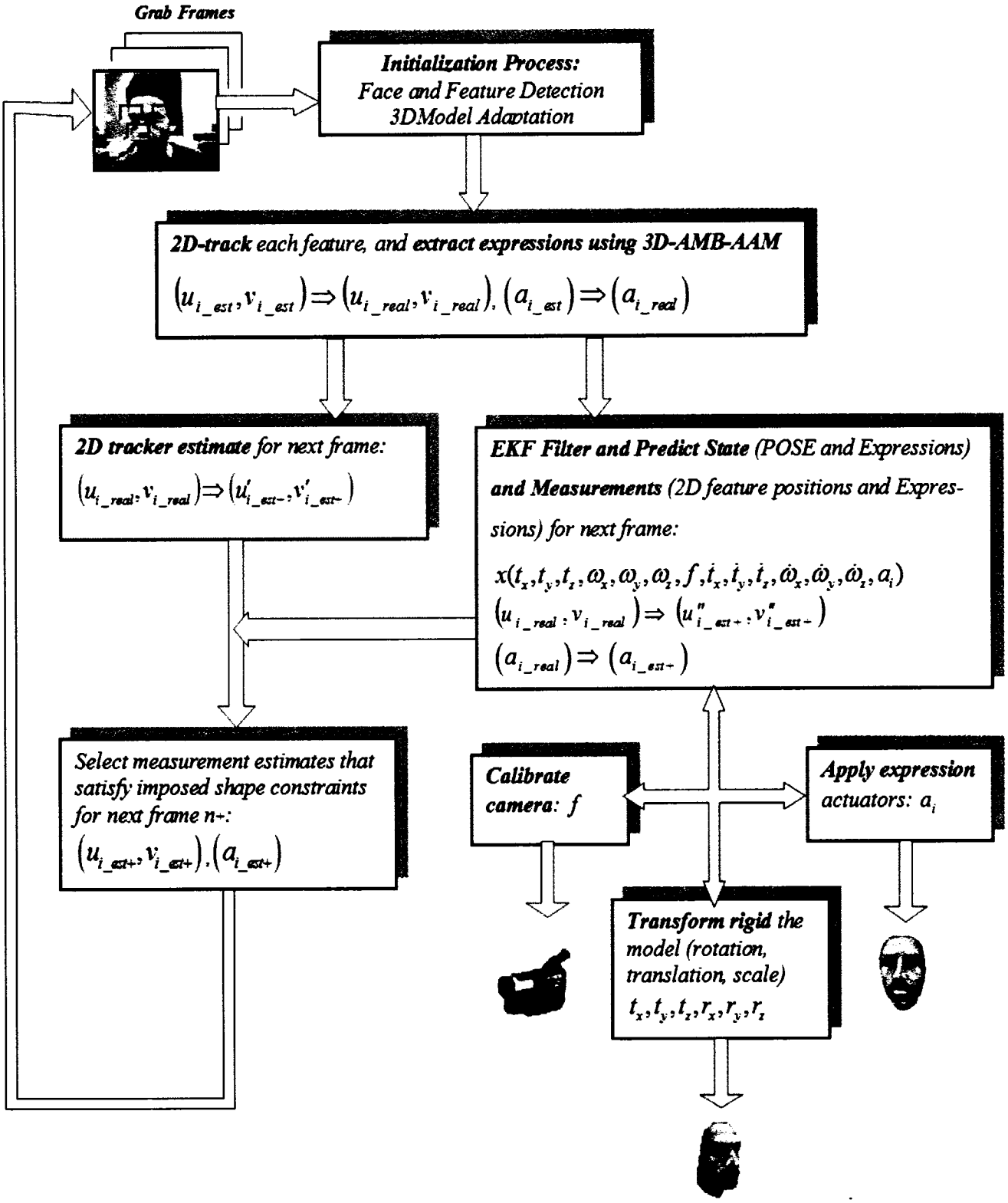


Fig. 5.3: Continuous 3D pose and expression recovery using an Extended Kalman Filter.

Tracking Recovery

Initialization and recovery are very important features in tracking applications. Large out-of-plane head rotations lead to tracking failures. The 2D tracking fails in case of accumulation of errors generated by measurement noise or mismatched points. These events ask for an automatic recovery algorithm to ensure tracking robustness and continuity. Face tracking systems that rely on incremental motion estimation with no recovery methods can successfully track only for short time in image sequences. The 3D structure of the feature points can help the tracking. The 3D head model is used to decide whether or not a feature is occluded. When features become occluded, they are assigned an infinite variance in the noise measurement matrix R of the Kalman filter. When large in depth rotations occur the tracked points become occluded and should be discarded from the tracking set. In this case, new known rigid features (such as ear, jaw points etc.) can be added to the EKF. In order to avoid matrix resizing each time new points are added, it is easier to include the new measurements with infinite variance from the start [Str99]. Every time the points become active the filter includes these measurements in the solution. When the points disappear they are put on hold until they reappear. The tracking is considered lost when the 2D KLT tracker signals that un-occluded features are not anymore reliable for tracking. Also, when fitting error of the ABS module is high for several frames is a strong indication that the pose tracking has failed and the expressions can no longer be found. In these situations, the detection process takes over, and the EKF is re-initialized.

Model Choice

Transforming a generic model into a personal one is a common step used in the initialization stage of model-based tracking. In the scenario of using an AAM to track faces, this transformation translates in representing the texture, facial types and facial expressions separately [Ahl03]. The active model should be adapted to the face by varying all parameters at the beginning. This way the active model is converted into a personal one, which will be controlled only by the expression parameters for the rest of tracking. This simplification is supposed to increase the speed of tracking since the computation time grows approximately linearly with the number of control c parameters used (texture and shape combined). One might argue that using more c parameters for an increased accuracy will slow down the fitting process. However, our

benchmarking shows that the real burden in ABS fitting is the *analysis* step, namely the *Sample Shape ( )* function. For instance, for an ABS iteration *Texture Instance ( )* took 0.22 ms, *Shape Instance ( )* took only 0.09 ms, and *Sample Shape ( )* completed in 5.58 ms. In our implementation *Sample Shape ( )* draws the textured and animated 3D model using OpenGL graphics hardware. *Sample Shape ( )* is also independent of control parameters, so that representing the model with a small number of control parameters does not increase the fitting speed. For this reason and for the purpose of an increased fitting accuracy we chose the combined model parameterization for tracking expressions. This will increase the generality of the active model, allowing for better handling changes during tracking, such as lighting variations. Moreover, the initial fit might not be accurate and keeping all control parameters (texture, facial types and expressions) permits updating the facial texture or geometry during tracking. It was shown in Section 2.3.4 that discarding the highest variance modes from the 3D AMB AAM formulation can lead to better fitting accuracy. The first modes of variation encode the highest variations across the training set, which is provoked mainly by lighting changes and extreme expressions. In our tracking experiments we discard the first variation mode from the texture formulation and the first two variation modes from the combined formulation. This way we keep the most discriminative modes for facial expression tracking.

Tracking Speed

The main question here is whether the algorithms presented in this thesis can be implemented in a functional real-time MBVC. A minimal NTSC frame rate of 15 Hz, allows 80 ms of processing per frame. Encoding expression parameters and updating facial texture can take less than 10 ms leaving the rest for feature tracking, speech synchronization, and other tasks. At this stage our tracking system performs at close to 10 Hz, without using optimizations. The tracking speed can be improved using:

1. specialized hardware for 3D graphics that includes texture mapping
2. a multi-resolution (coarse-to-fine) AAM, which is known to improve also the fitting accuracy
3. reducing the number of iterations of the model fitting algorithm

5.3. EXPERIMENTS

The performance of the 3D pose tracking module in a synthetic sequence was demonstrated in [Cor02]. We validate our new tracking system with three sets of experiments. The experiments illustrate the performance of the tracker with synthetic and real sequences of a human face moving freely in front of the camera and performing different expressions.

Similarly to [Cor02] the first set of experiments performs a calibration of the tracking system, using a synthetic sequence of 3D face model performing rigid and non-rigid motion (*synthetic-pose-expression experiments*).

The second set of experiments uses real image sequences from MMI database [Pan05], where faces are displaying different expressions but have limited rigid motion (*real-expression experiments*).

The third set of experiments is performed in the BU real image sequence database [Sca98], which contains faces with a large range of rigid motion. The images of the sequence were captured at the resolution of 320x240 pixels in different lighting conditions. We choose a subset of this database (four movie files) where persons perform facial expressions (jaw drop and smile) as well (*real-pose-expression experiments*).

For tracking experiments we decided to build a different 3D AMB AAM. This time the model was trained to recognize four individual EAUs (Jaw Drop, Mouth Corners, Eyebrow Middle, and Eyebrow Inner). The model becomes more flexible, and is capable of generating different spontaneous expressions including the prototypic ones. The trained 3D AMB AAM model is described by 9 shape modes, 76 texture modes and 32 combined modes that retain 92 percent of the combined shape and texture variation of the training set of faces. The normalized training images had a resolution of 128 by 128 pixels, containing a facial texture patch of approximately 2200 pixels. The normalized framework was defined by the landmarks *n* (*nasion,* skull feature between the eyebrows) and *sn* (*subnasale*, the center of the nose base), which define the anthropometric vertical axis (Section 2.3.2).

5.3.1. SYNTHETIC-POSE-EXPRESSION EXPERIMENTS

In order to validate the accuracy of our 3D head tracking system we built a synthetic image sequence. A previously recorded sequence of 2D images representing 3D-head model poses and expressions is played as "live" video, and tracked with our EKF system. The estimated motion values $(t_x^e, t_y^e, t_z^e, \theta_x^e, \theta_y^e, \theta_z^e, a_i^e)$ are compared with the measured motion values $(t_x^m, t_y^m, t_z^m, \theta_x^m, \theta_y^m, \theta_z^m, a_i^m)$ of the synthetic image sequence. In the above representation (t_x, t_y, t_z) is the 3D position, $(\theta_x, \theta_y, \theta_z)$ is the 3D orientation of the head, and the $a_i, i = 1,2,3,4$ represent the four tracked EAU (Jaw Drop, Mouth Corners, Eyebrow Middle, Eyebrow Inner). We measured the performance of orientation and expression recovery. The played sequence of 225 frames displays a synthetic head performing large rotations in range $(-45, 45\,\mathrm{deg})$ for "Yaw" and $(-20, 20\,\mathrm{deg})$ for "Roll" and "Pitch" angles. Fig. 5.4 shows the flowchart of a calibration process using a computer generated sequence pilot.

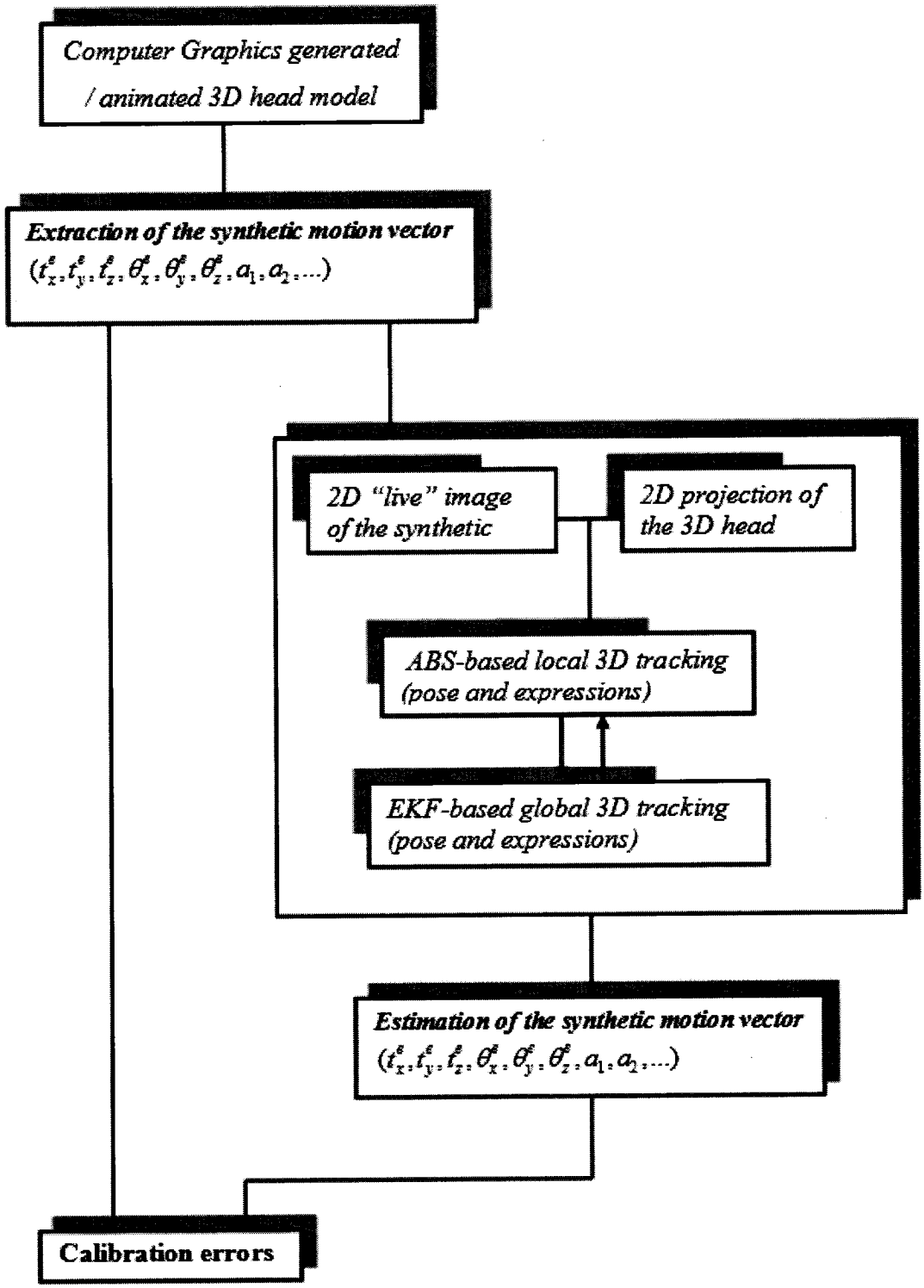


Fig. 5.4: Tracking calibration flowchart.

The resulting errors are the consequences of 3D/2D initial point-identification and 3D rigid and non-rigid tracking. We minimized the errors by fine-tuning the initialization process of the EKF. The crucial EKF parameters for good performance are the noise covariance matrices R and Q (Section 4.3). The state covariance matrix Q relates to input and model errors, representing prior knowledge about the state vector, and R relates to the measurement errors. Tuning R to small values will emphasize on the measurements making the filter "jumpy". Setting the initial Q too small also tends to make the filter instable. Tuning R large and Q small will render the filter slow, incapable of keeping the pace with fast changes in the state vector. We start with approximate, large enough Q and R, and then use trial and error to obtain maximum tracker performance. During the calibration process we evaluate the tracker accuracy by computing the root mean square error (RMSE) between true and estimated rotation angles, and between true and estimated expressions (EAUs):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{true} - y_{estim})}, \tag{5.1}$$

where y_{true}, y_{estim}, are the ground-truth and estimated angle or EAUs parameters, and n is the number of frames in the sequence. We found experimentally in one case that the calibrated RMSE for rotation angles are:

$RMSE\_X = 3.395$ (Pitch)
$RMSE\_Y = 4.565$ (Yaw)
$RMSE\_Z = 2.552$ (Roll)

The resulted calibrated RMSE for expressions (EAUs) are:

$RMSE\_JD = 0.095$ (Jaw Drop)
$RMSE\_MC = 0.122$ (Mouth Corners)
$RMSE\_EM = 0.256$ (Eyebrow Middle)
$RMSE\_EI = 0.105$ (Eyebrow Inner)

Fig 5.5 shows the 3D-tracking of a synthetic head, and Fig. 5.6 shows the *estimated* vs. *real* 3D orientation for a calibrated synthetic sequence. These statistics are comparable to the Polhemus sensor accuracy [Aza95] indicating that the vision estimate can be accurate as the Polhemus sensor. In some frames the model fitting (expression) was less accurate due to significant pose change, and occlusion.

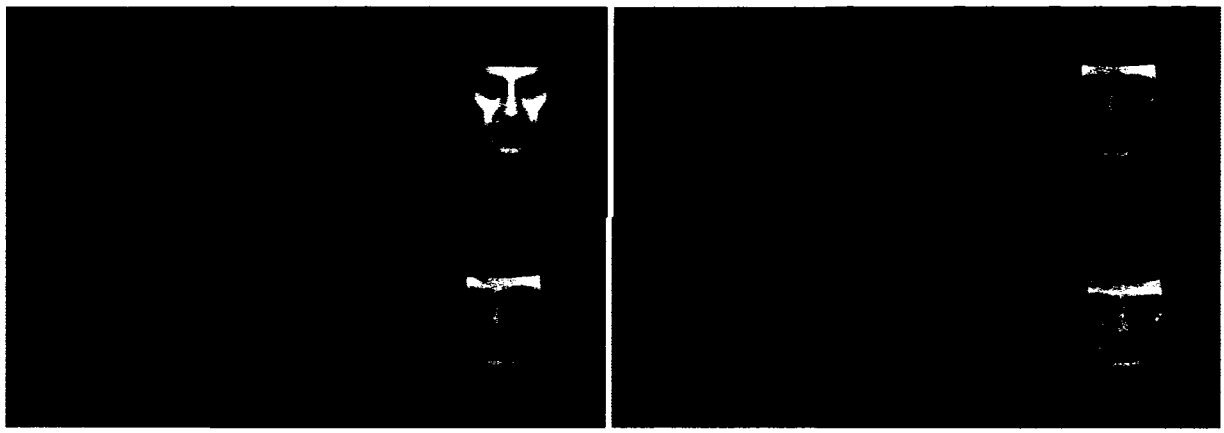


Fig. 5.5: Tracking instances from the synthetic-pose-expression sequence (original, normalized, and synthetic frames). The face is from MMI database [Pan05]).

Fig 5.7 shows the *estimated* vs. *real* animation of three facial expressions (Jaw Drop, Mouth Corners, and Eyebrow Inner) for a calibrated sequence. The experiment showed that the accuracy of expressions tracking increased when the face is closed to frontal position. The expression tracking drifted for extreme rotation angles that cause reduction of expression information. Our experiments also showed that the RMSE of each expression is lower if we do not take in consideration the first few frames necessary for the EKF to converge. Fig. 5.8 illustrates snapshots of head tracking in a synthetic-pose-expression sequence. For each row the left side shows the synthetic head and the right side represents the result of tracking that is a 3D head mimicking the estimated rigid and non-rigid motion.

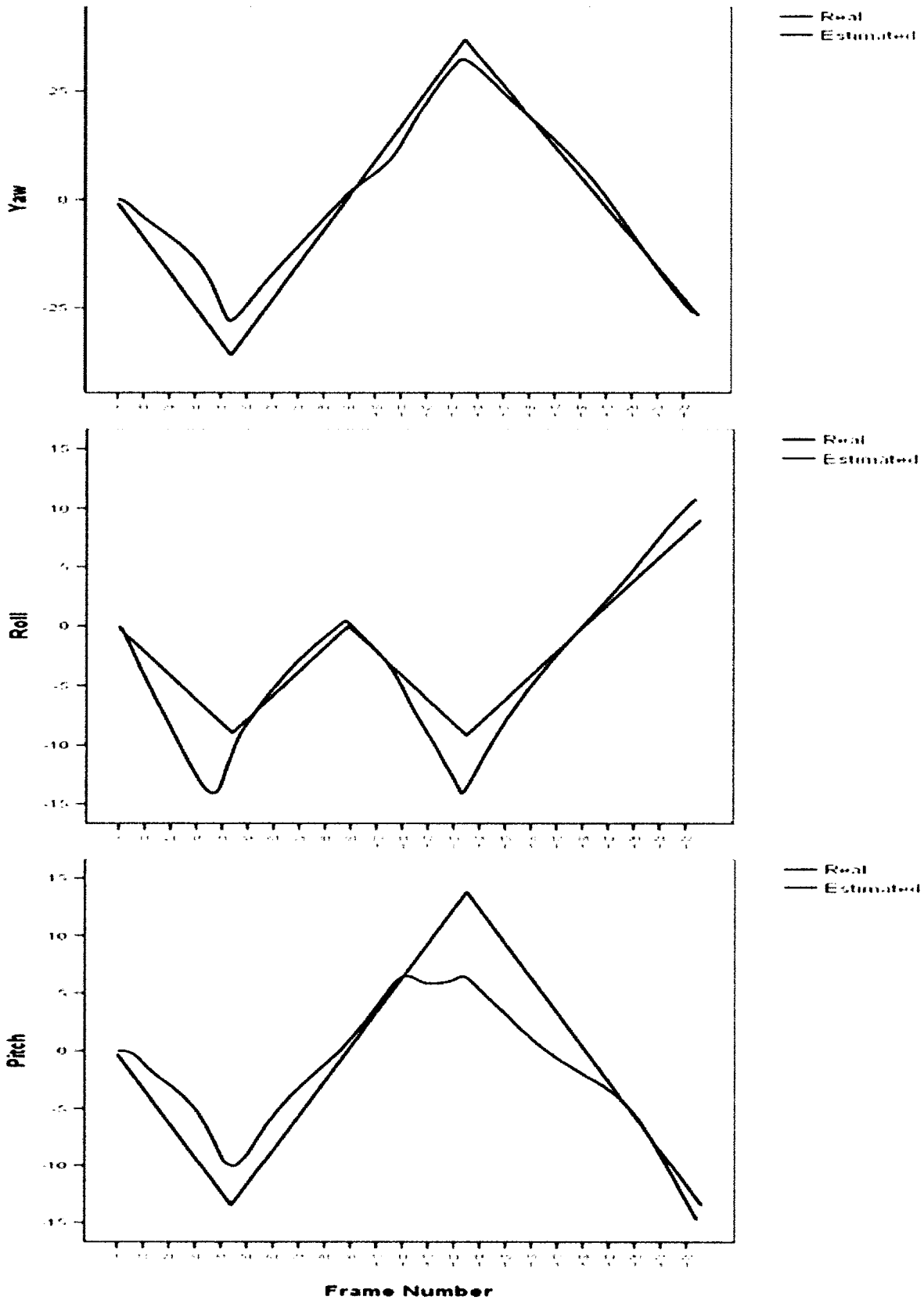


Fig. 5.6: *Real vs. Estimated* orientation for a synthetic pose-expression experiment.

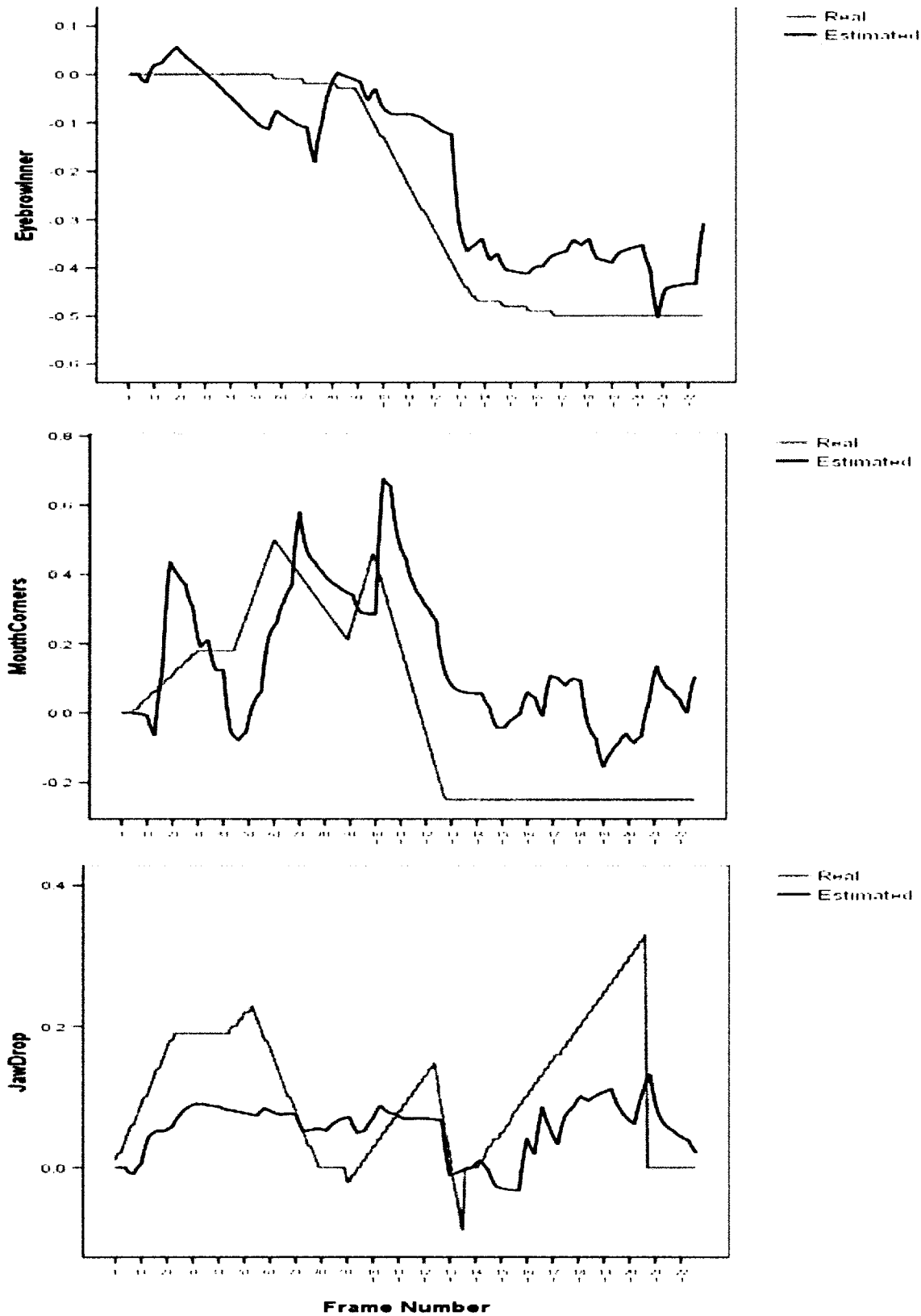


Fig. 5.7: *Real vs. Estimated* expressions for a synthetic pose-expression experiment.

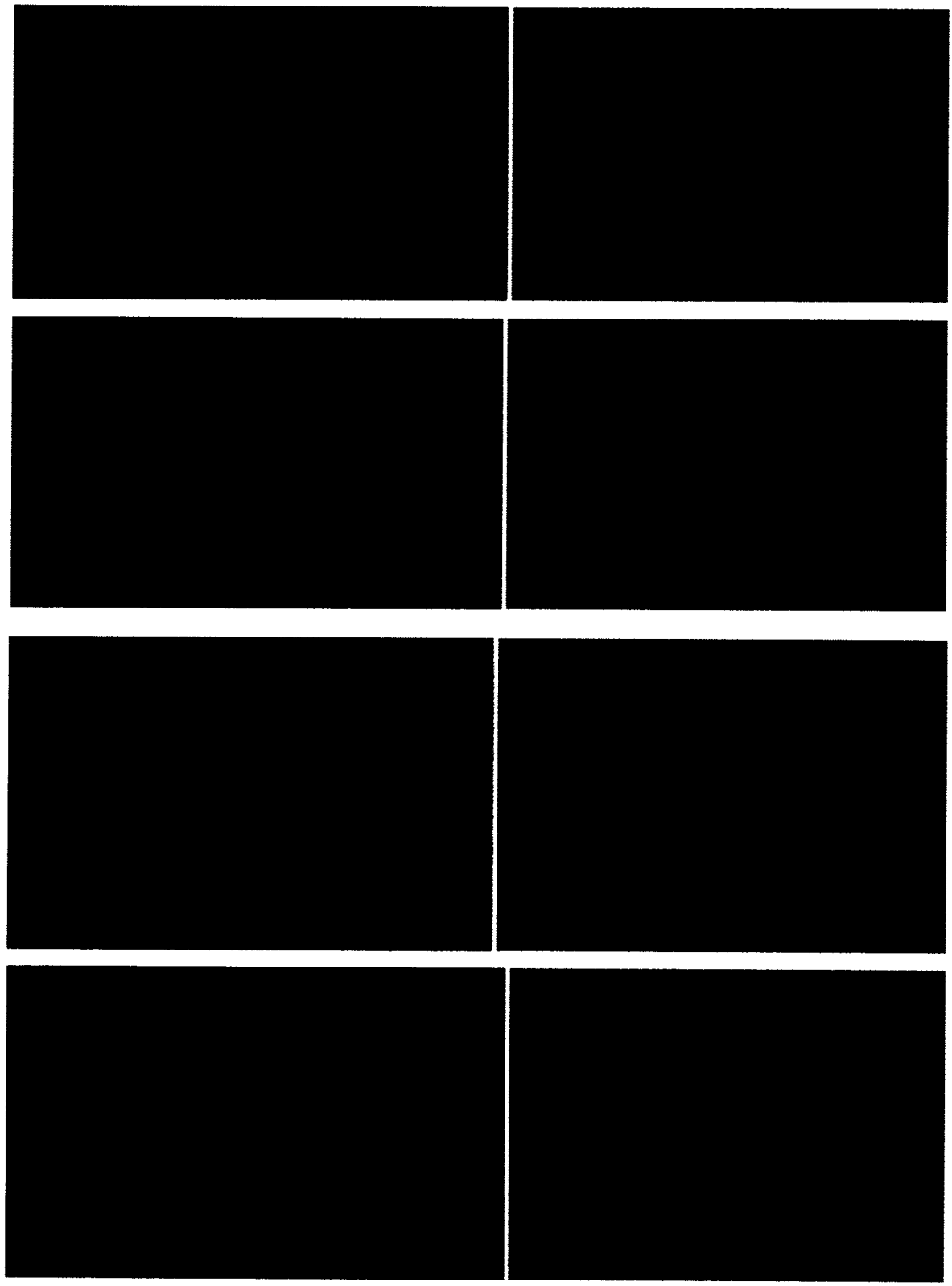


Fig. 5.8: Head tracking in a synthetic-pose-expression sequence (face from [Pan05]).

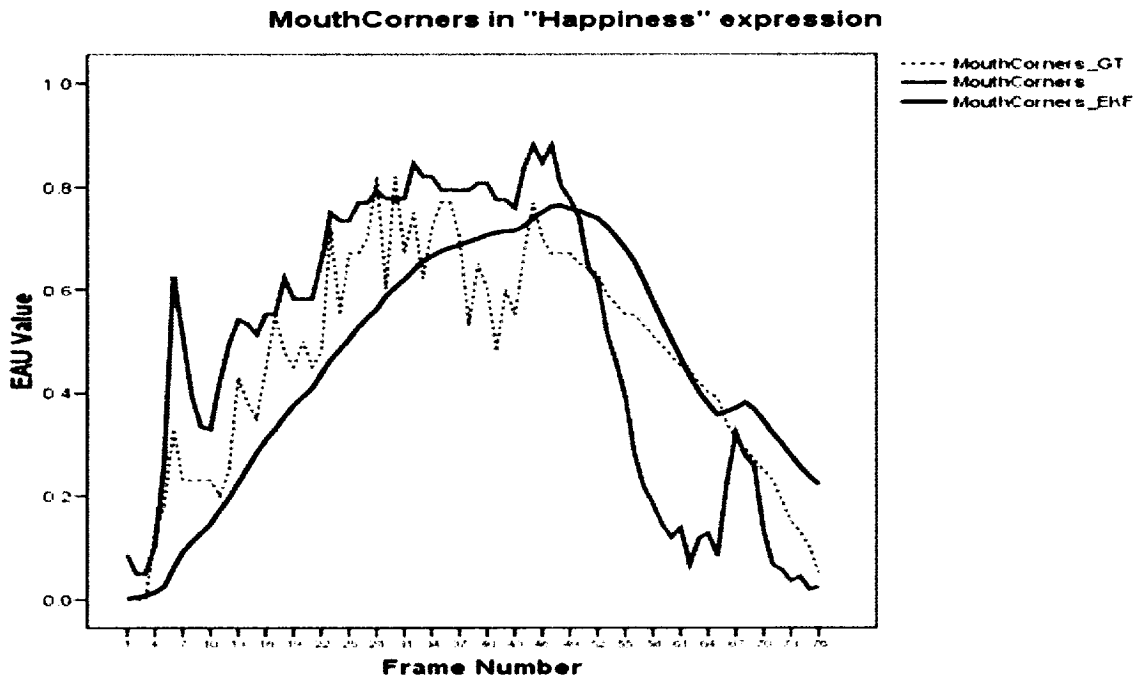
5.3.2. REAL-EXPRESSION EXPERIMENTS

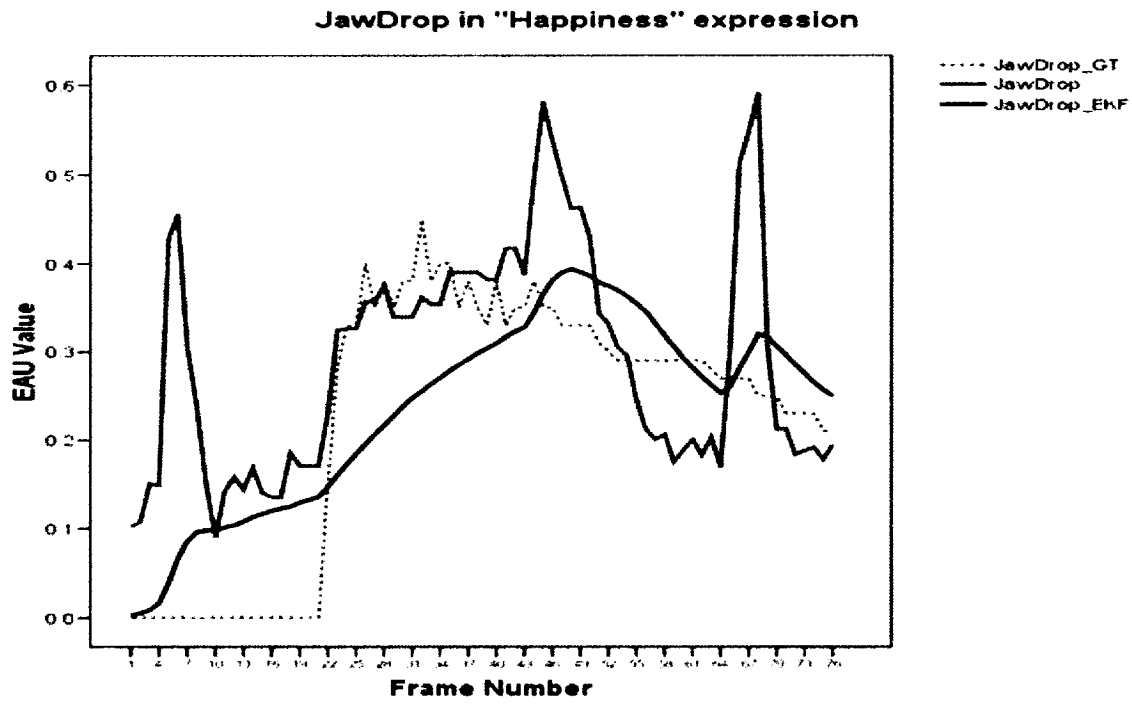
Image sequences from the MMI database [Pan05] expose frontal or near-to-frontal faces. The faces display emotional expressions starting from the neutral state. Each video sequence was used as the input to the face tracking algorithm. The sampling rate was 25Hz, and the length of each video-expression is between 20 and 200 frames. In this experiment the ABS module fits 3D AMB AAM on each image of a video sequence, using the estimated EKF output as initialization of appearance and pose parameters.

Fig. 5.10 illustrates a typical tracking sequence of a person present in the training database. The first icon of each image pair shows the normalized shape-free image. The second icon shows the synthesized face overlying the real input image. The frame number, the number of iterations taken by the fitting process and the corresponding matching error are also displayed. In order to study the effect of including the deformation parameters in EKF we conducted two types of experiments in each sequence.

In the first experiment we tracked a face using only the 3D AMB AAM, by sequentially fitting the active model in each frame. At each frame the ABS iterative search starts with the previous frame estimate of model state (shape-appearance parameters). The pose parameters are provided by the EKF, given its proven stability with large pose variations and the fact that the active model was trained to handle only small pose increments. An ABS tracking system includes a feedforward component, but it lacks a model of system dynamics. This aspect and the weak stability over time of the gradient descent search algorithm will soon lead to failure.

The second experiment included the animation parameters in the recursive framework. The great benefit of using a recursive estimator in tracking a facial expression is illustrated Fig. 5.9. The dashed gray line represents the ground-truth of the AU throughout the sequence. The light-colored line shows the output animation parameter synthesized by the fitting process without using the EKF. The dark line shows the "smoothing" effect of the EKF over the expression parameters. Fig. 5.9 also illustrates the statistics of tracked expressions "with" or "without" the presence of the EKF.





Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
JawDrop	76	.09	.59	.2918	.12824
JawDrop_EKF	76	.00	.39	.2355	.10866
MouthCorners	76	.02	.88	.4887	.29038
MouthCorners_EKF	76	.00	.76	.4560	.23095
Valid N (listwise)	76				

Fig. 5.9: The effect of EKF on expression tracking for a real-expression experiment.

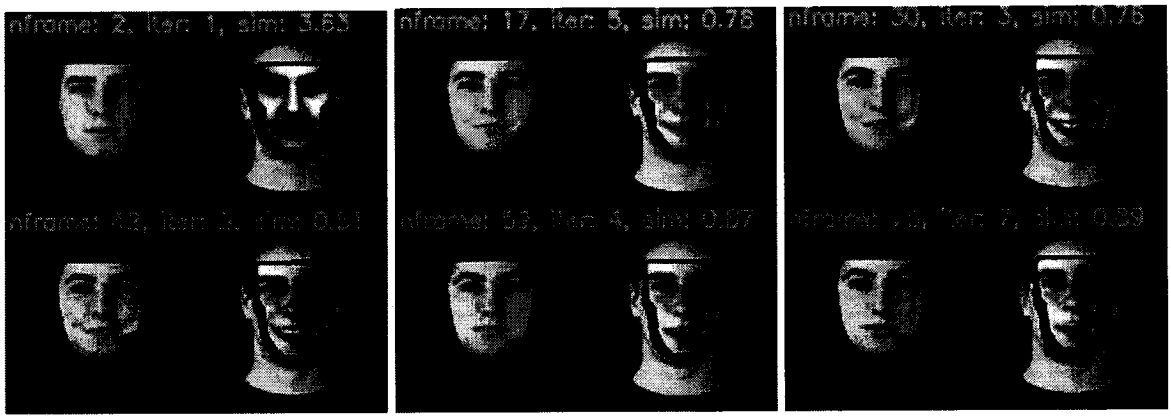


Fig. 5.10: A typical real-expression tracking sequence (face from MMI database [Pan05]).

We tested the system in ten real-expression tracking sequences. Tracking performed well in all image sequences, both the rigid and non-rigid modules did not fail. This is because the faces have frontal orientation, limited rigid motion and no occlusion, which allowed both tracking methods to follow the expression signature without failing. Two of the sequences were AU ground-truth annotated using the projection framework presented in Section 2.2.3. For the rest of the sequences the judgment of the tracking performance was subjective.

5.3.3. REAL-POSE-EXPRESSION EXPERIMENTS

Image sequences from [Sca98] database contain faces performing large rigid motions in different illumination conditions. Some faces display spontaneous expressions. No face was present in the training set. We selected a subset of four pose-expression videos as the input to the face tracking algorithm. The sampling rate was 15Hz, and the length of each pose-expression video is over 400 frames (8-15 seconds long). Also, the pose ground-truth throughout the image sequences was collected with a 3D "Flock of Birds" sensor [Flo00].

The tracking system can cope with different degrees of rigid motion and non-rigid deformations as illustrated in Fig. 5.11. The estimated orientation during tracking against the ground-truth for the displayed example is illustrated in Fig.5.12. The effect of the EKF in non-rigid tracking is dramatic in this case. As seen in Fig. 5.13 the tracking of facial expressions would have been difficult without using the Kalman filter. The extreme values illustrated by the graph in light

color indicate ABS tracking failures, when no EKF was used. The low performance of the 3D AMB AAM on constantly fitting the face without the help of EKF might be caused by two facts: the face was not present in the training dataset, and the large range of rigid motion. Since no ground-truth for expressions was provided, the tracking accuracy was judged subjectively. For the illustrated example in Fig 5.11, the subject displays a "happiness" expression starting in the second half of the sequence (i.e. the mouth and eyebrow corresponding AUs were activated).

Fig. 5.11: A typical real-pose-expression tracking sequence (face from BU database [Sca98])

In some frames the model fitting is less accurate due to significant pose and/or expression change, or simply the absence of that particular facial instance in the training set. However, the smoothing effect of Kalman filter makes the fitting process stable and robust over time despite errors in individual images. Tracking performed well in all four image sequences. Despite large rigid motions, rigid tracking did not fail. Based on residual error the non-rigid tracking drifted away between two and four times in two sequences, mainly due to the pose changes, but recovered quickly with the help of the EKF pose and expression estimations. The non-rigid tracking was considered lost when its residual error had a high value for several frames. In case of non-rigid tracking failure, the ABS search discarded the estimated model parameters and started with the EKF estimated pose and the mean of model parameters.

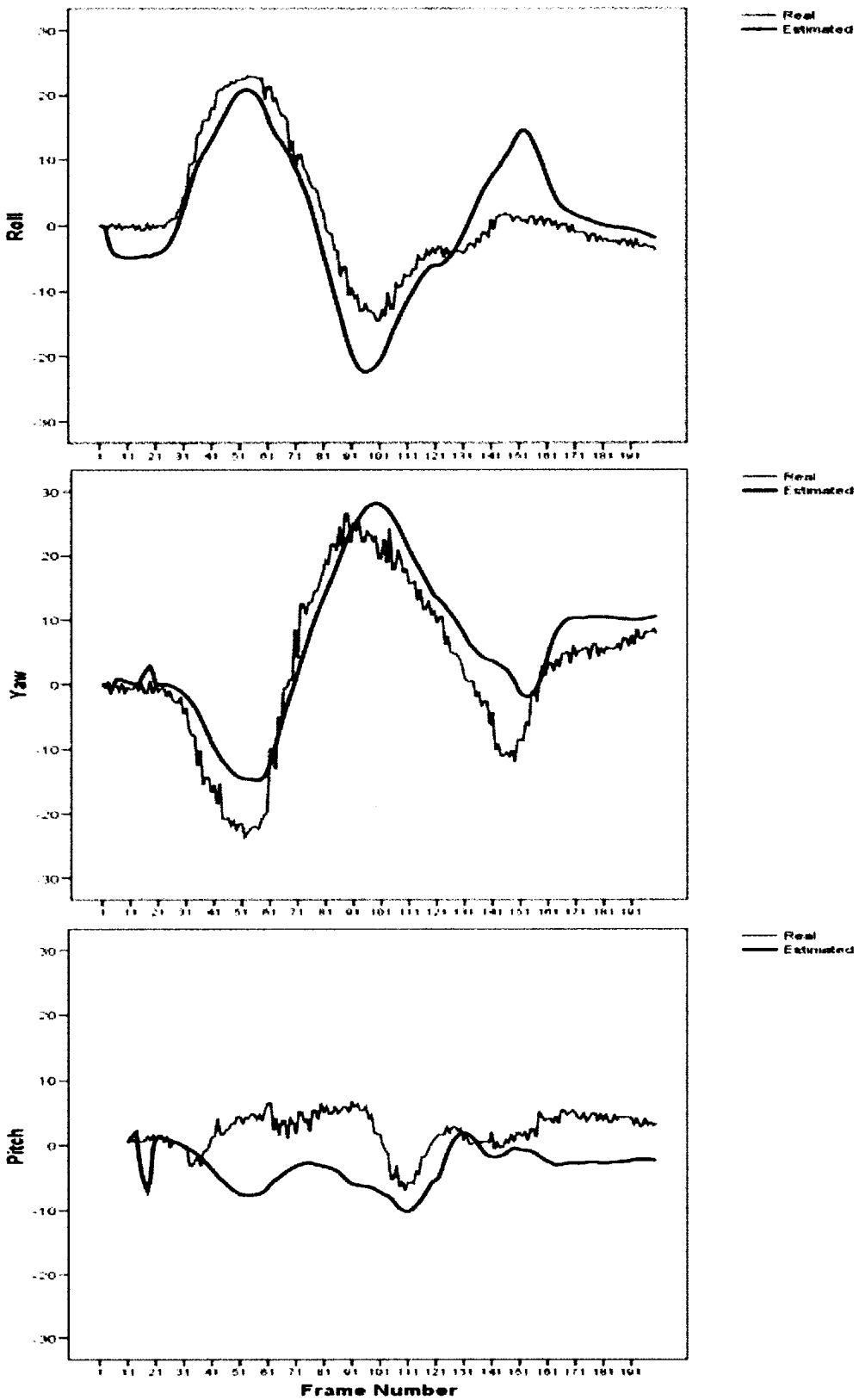
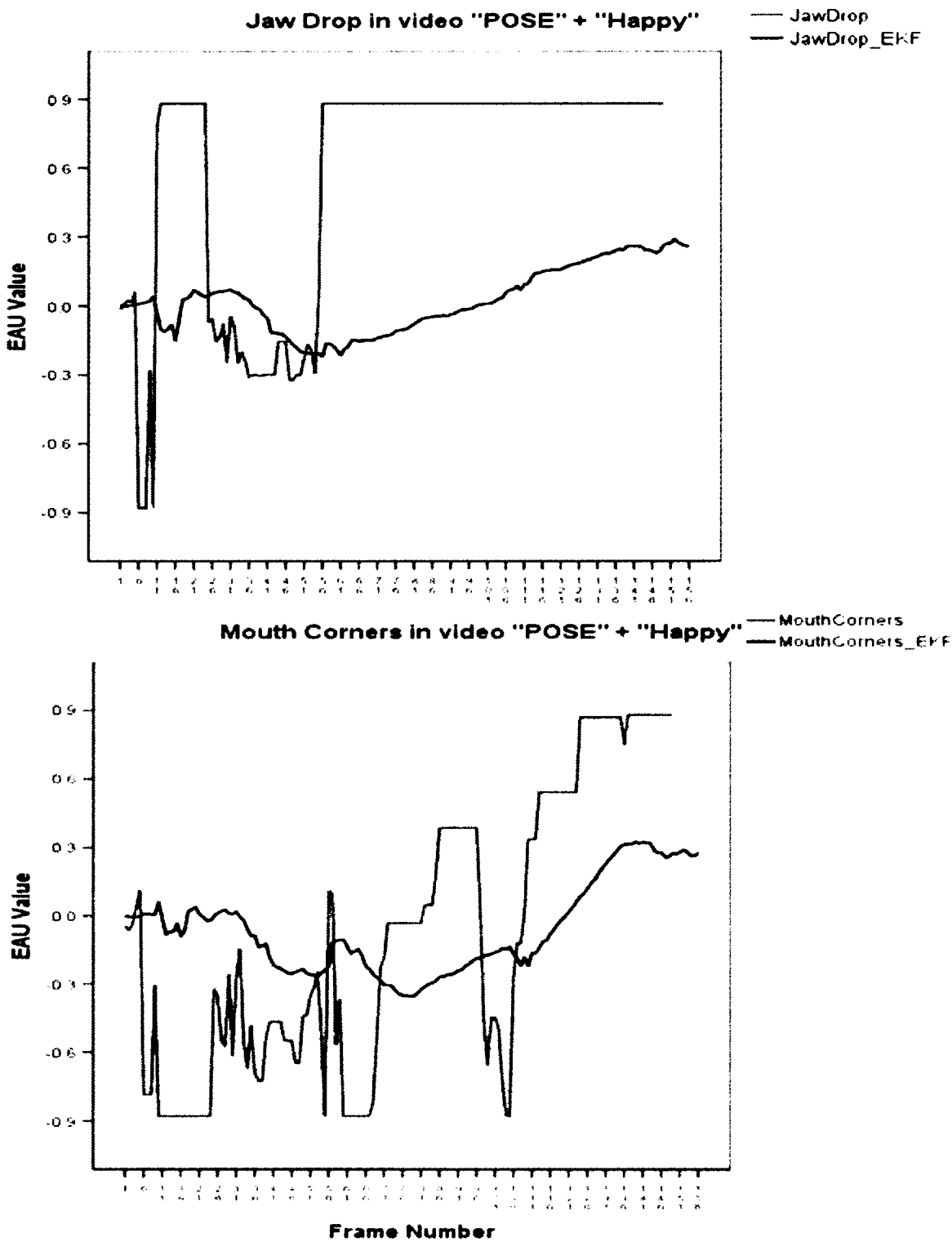
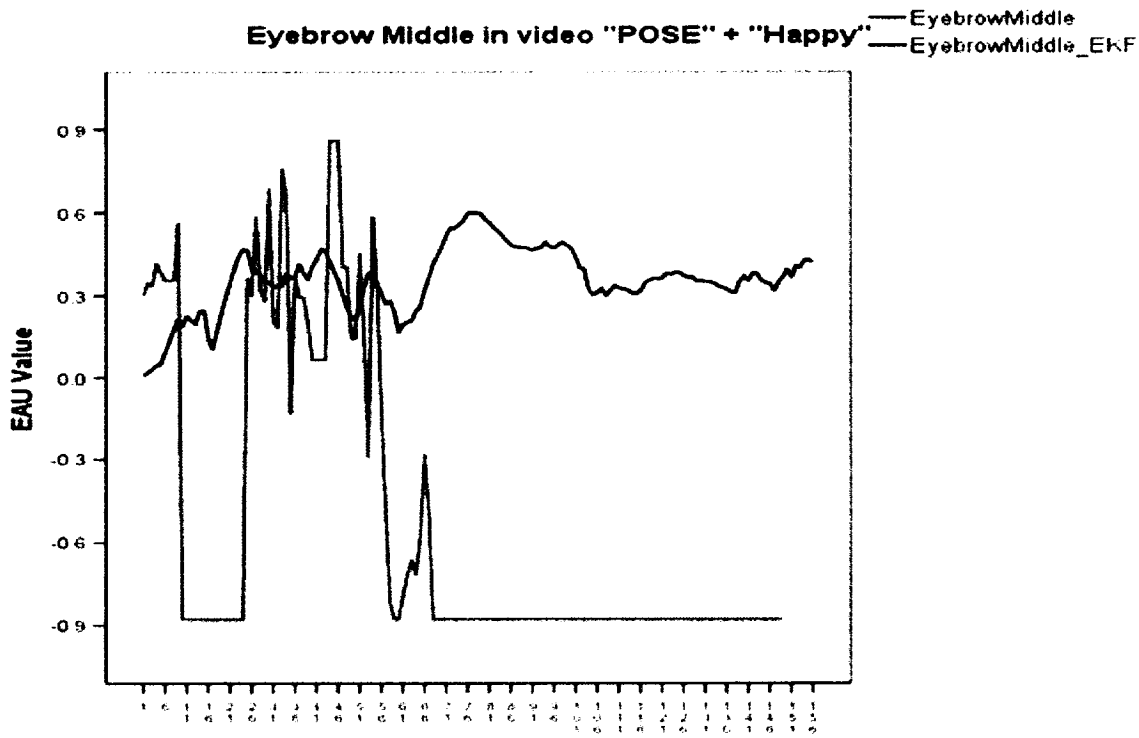
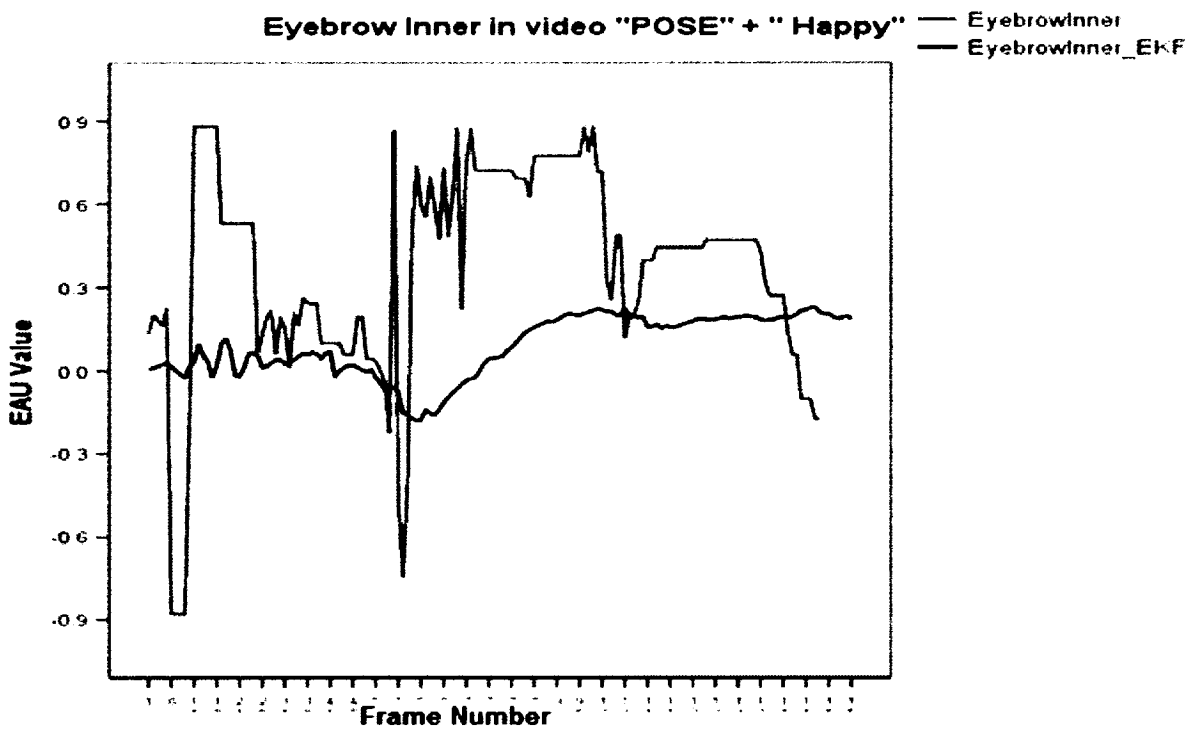


Fig. 5.12: *Real vs. Estimated* orientation for a real-pose-expression experiment.


Jaw Drop in video "POSE" + "Happy"
JawDrop
JawDrop_EKF
EAU Value
0.9
0.6
0.3
0.0
-0.3
-0.6
-0.9
Mouth Corners in video "POSE" + "Happy"
MouthCorners
MouthCorners_EKF
EAU Value
0.9
0.6
0.3
0.0
-0.3
-0.6
-0.9
Frame Number


Eyebrow Middle in video "POSE" + "Happy"
EyebrowMiddle
EyebrowMiddle_EKF
EAU Value
0.9
0.6
0.3
0.0
-0.3
-0.6
-0.9


Eyebrow Inner in video "POSE" + " Happy"
EyebrowInner
EyebrowInner_EKF
EAU Value
0.9
0.6
0.3
0.0
-0.3
-0.6
-0.9
Frame Number

Descriptive Statistics

	N	Minimum	Maximum	Mean	Std. Deviation
JawDrop	149	-.88	.88	.5685	.52111
JawDrop_EKF	156	-.22	.29	.0245	.14405
MouthCorners	149	-.88	.88	-.0849	.62201
MouthCorners_EKF	156	-.36	.32	-.0615	.19732
EyebrowMiddle	149	-.88	.86	-.5297	.55661
EyebrowMiddle_EKF	156	.01	.60	.3566	.11978
EyebrowInner	149	-.88	.88	.3694	.37765
EyebrowInner_EKF	156	-.18	.23	.0863	.11016
Valid N (listwise)	149				

Fig. 5.13: The effect of EKF on expression tracking for a real-pose-expression experiment.

The tracking system was also tested in a light varying image sequence. From 70th to the 120th frame a strong shadow was cast from the right side of the subject. During this period the ABS process had problems matching the face (i.e. finding the facial expression) in three instances. However the rigid tracking did not fail due to the robustness of the EKF. Every time the non-rigid tracking had the tendency to drift, it started the next frame from the correct position attributable to the EKF prediction. One such example is illustrated in Fig.5.14. The non-rigid tracking was lost on frame 83 and successfully recovered on the next frame.

Fig. 5.14: Tracking a face in light-varying scene (face from BU database [Sca98])

5.4. AN AUTOMATIC FACE TRACKING SYSTEM

The "Automatic Face Tracking System" (AFTS) was implemented in a modular fashion, in order to facilitate experimentation with a variety of tracking and animation methods (Fig. 5.15). When the tracking system is used in *Videophone* (Performance-Driven) mode, deformation parameters are received automatically. If used in *Annotation* (Scripting) mode, the system can be used to annotate a database of faces that is to extract parameters (pose, shape and texture) for AAM training (Chapter 2). The AFTS consists of two main modules (Fig. 5.15):

1. *Animation/Rendering Module*
2. *Modeling/Control Module*

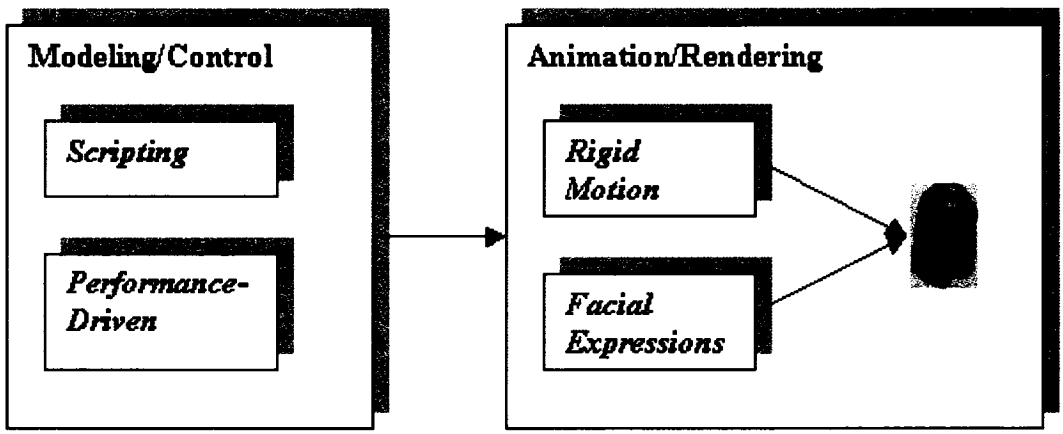


Fig. 5.15: The architecture of the Automatic Face Tracking System.

5.4.1. ANIMATION/RENDERING MODULE

A "baseline" state of the facial model corresponds to a face in a neutral position, where simulated muscles are mapped to the wireframe model in a relaxed state. During the animation, a series of muscle induced deformations are applied to the facial model points in the baseline position, after which the face is rendered. Eyes are constructed from spheres with iris and pupil. The eyes rotate simultaneously in their coordinate system. Eyelids are wireframe models, which can blink automatically during the animation. Two types of "controllers" control the whole facial deformation at the low level:

1. *Muscle* controllers: the 14 muscle parameters using *muscle-based* model, which are mapped to AUs.
2. *Jaw, Eyelids*, and *Eyes* controllers: 3 parameters using *key-node parameterization* model

The animation module can also control the rigid motion of the human head, through basic transformations (translation, rotation and scaling) of the 3D-head frame. The object-oriented implementation of the system permits to add new controllers in the future.

5.4.2. MODELING/CONTROL MODULE

The Modeling/Control Module contains two blocks that allows for a textual scripting description of the animation program or for a gesture-driven animation:

1. Scripting Block
2. Gesture/Performance Block

Scripting

The Scripting block was conceived as an open interface, which enables different animation modules to be easily added. Currently the scripting block has a low-level implementation, which allows generating facial deformation in three modes:

1. *EAU* macro controllers: the 4 implemented combined AU muscle activations (*Jaw Drop, Mouth Corners, Eyebrow Middle,* and *Eyebrow Inner*), used in this research.
2. *EAU* emotional controllers: the 6 implemented prototypic expression using muscle-based model (*Surprise, Happiness, Anger, Fear, Sadness,* and *Disgust*).
3. *AAU* controllers: the 11 anthropometric AUs (*Cranial Width, Face Width, Face Height,* etc.).

These low-level parameterizations provide a basis for a future high level of programming and scripting techniques:

1. *Speech-Driven Level:* animating directly from a speech soundtrack
2. *Behavior-Driven Level:* expressing the desired actions in terms of "behaviors"
3. *Story-Driven Level:* based on major activities such as: story understanding, stage direction, and action generation.

Performance-Driven

Generally, Performance-Driven animation involves measuring real human actions to drive synthetic characters. Data used to drive the animation comes from interactive input devices such as: data gloves, instrumented body suits, vision-based motion system, and other various sensors. Since our goal was to construct a real-time, low-cost videoconferencing system, the performance-driven control was implemented as a vision-based module, which is the core of the tracking system.

The tracking system was implemented using C++ under Windows environment. The current implementation runs at 10 fps on a PC with a 2.3 GHz Pentium 4 CPU and an ATI Mobility Radeon 9700 graphics card. The recovered 3D orientation and expressions are propagated to the Animation Module, which renders a new posture of the 3D model (Fig. 5.16).
During the initialization phase when the 3D active model needs to adapt the facial image, we allow a maximum number of iterations of 8. During tracking the initial estimate is closed to the optimum, hence we observed that less than 8 iterations are needed to fit the active model on each frame. Based on this, we allow only 4 iterations for model fitting during tracking. In this implementation the active model search needs about 25 ms per iteration that is the 3D AMB AAM fitting algorithm takes approximately 0.1 seconds for each frame.

Fig. 5.16: Tracking different facial expressions with AFTS (faces from MMI database [Pan05]).

Chapter 6

Conclusions and Further Development

6.1. CONCLUSIONS

Visual communications over low bandwidth (below 10 Kbps) channels is possible only if implementing high efficiency coding methods. One such technique is MBVC. MBVC has emerged as a very low bit rate video compression method suitable for videoconferencing and videophone applications. A priori knowledge of the scene contents can help representing objects by synthetic models. For example the contents of a video telephony stream are known a priori that is sequences of head-and-shoulder images of a talking person. MBVC uses this semantic information can be used to encode the scene with high efficiency. This means that the MBVC increases coding efficiency by using knowledge about the scene content and describing the real world geometry by 3D model objects (e.g. 3D face models). The main advantages of MBVC over the classical interframe, block-based compression methods such as MPEG 1/2 and H.261/3 are:

1. The low transmission rate needed for communication and low storage costs.
2. The reduced bandwidth requirements allow for other processing tasks such as face stabilization.
3. The model used at the receiver may be different from the model used at the transmitter.
4. The number of motion and model parameters to transmit over the communication channel does not change if the sender or receiver changes the image resolution.
5. The number of motion and model parameters does not depend on the size of the encoded objects.
6. No object distortions occur in case of tracking failures.

These advantages made MBVC a hot research topic for more than three decades.

This main goal of this thesis was the development of an Automatic Face Tracking System (AFTS) that can simulate the sender or receiver end of a Model-Based Video Coder. The most sophisticated module is the sender (encoder) module because of the face analysis and tracking

algorithms that have to be implemented. AFTS can track the pose and expressions of faces in a video sequence. The non-rigid animation parameters are encoded using muscle-based face animation techniques. The system works in real-time at 10 fps, and the experimental results look promising. With further development and optimization, a 30 fps real-time encoder that is a 3D face pose and expression tracker should be possible to implement on consumer hardware. The goal of the developed framework is to recover both head-pose and facial expressions from the performer facial movement. The effects of head motion and facial expressions will be non-linear coupled in a 2D video sequence, so that a successful MBVC method will have to separate the rigid from the non-rigid motion.

An important step toward the proposed goal was the development of a 3D AAM. The 3D model is based on muscle actuators to model facial expressions and anthropometrical controls to model facial types. Due to its increased generality, compactness, specificity, and parameter control our method is useful in real applications using face synthesis and analysis. The main advantage of formulating the active model in AE space is that the anthropometric characteristics and facial expressions are obtained directly during the synthesis phase. The classical 2D AAM recovers the point positions of the 2D shape without the knowledge of the underlying facial type or expressions. We have compared our algorithm with the classic AAM algorithm and have demonstrated its superior performance in image segmentation applications. Also, our model was intensively used in experiments to recognize facial expressions in FACS-standard images.

The second contribution of this thesis is a new 3D tracking algorithm allowing real-time recovery of 3D position, orientation and facial expressions of a moving head. The system can track four EAUs (Jaw Drop, Mouth Corners, Eyebrow Middle, and Eyebrow Inner), capable of generating different spontaneous expressions including the prototypic ones. The described method uses an EFK to extract the head pose (global motion) and the newly developed 3D AMB AAM to extract the facial expressions (local motion). The 3D AMB AAM fitting process is used to initialize the 3D head tracking system. The model provides better initial depth parameters to the EKF, and is used in the recovery process in case of tracking failure. The resulting motion tracking system was shown to work in a realistic environment without makeup on the face, with an uncalibrated camera, and unknown lighting conditions and background. The head tracking was successfully validated in synthetic and real standard image sequences.

6.2. FUTURE WORK

The main future work has to concentrate on improving the AFTS stability in the presence of difficult tracking conditions such as: illumination changes, occlusions, changes of viewpoint, and rapid motions. Although the 3D AMB AAM image interpretation and AFTS perform well, there is a number of pending issues as outlined below:

Illumination Problem

Usually, in uncontrolled environments the light may fluctuate significantly over time. Changes in illumination can drastically modify the intensity and shading of an image. The ABS expression estimation component of our tracking system, matches the synthetic facial image with the original one at each frame step. Estimating tracking parameters becomes difficult if differences between synthetic and original images are too large due to significant light variations. In other words, serious illumination inconsistency increases the tracker inability to adapt to changes of object appearance. As a result, specific illumination models are needed in order to control lighting changes in the scene.

The 3D-AMB-AAM does not explicitly model the light parameters. Instead, facial images captured in arbitrary light conditions were used to train the active model. This aspect and the robustness of feature tracking to light changes, and the recursive framework kept the tracker from drifting away from target. However, extreme light variations can cause the tracker to fail without the use of an illumination model. During the light varying sequence the AFTS lost the expression tracking, but recovered spontaneously. Even if the tracker may recover fast, methods for improving robustness in arbitrary light environments are needed.

Tracking Accuracy and Speed

The goal of the developed AFTS is to recover both head-pose and facial expressions from the performer facial movement. The accuracy of the tracker was measured in ground-truth synthetic and real pose-expression sequences. Even the results are very encouraging an increase in accuracy is needed especially in the area of expression recovery. In our synthetic sequence

exposing large rigid motions and spontaneous expressions the AFTS followed subject's non-rigid motion approximately (Fig. 5.7). The need of high quality expression tracking is obvious in videoconferencing and videophone applications where the speech soundtrack has to be synchronized with the mouth motion. Failures to track the mouth motion during speech create disturbing effects for human observers. Future work will concentrate in improving the tracking accuracy and speed especially for non-rigid motions. Also, for an increasing quality of decoded faces, real life videophone applications may require a blend of geometry-based texture warping and image-based interpolation.

The developed AFTS demonstrated the great value of including the facial expression tracking in the EKF recursive framework. Since our tracking framework filters and predicts muscle intensities, the facial expression model remains the researcher's choice. One direct and simpler variant of our 3D AMB AAM would replace the AAM by an ASM. ASM uses models of the image texture only in small regions around each landmark; hence it is faster than AAM. ASM was also shown to be more accurate than AAM in specific feature point location experiments [Coo99a], which makes it a good candidate for facial contour tracking. The derived 3D AMB ASM will use the projection of the 3D facial contours depicted from our wireframe face model. However, since AAM minimizes the texture errors, it can be used for facial texture updates and as a backup tracking component.

Texture Update and Quality of Reconstruction

AFTS extracts the subject texture from the first frame of the video sequence. A texture update scheme would increase the quality of texture mapping during tracking. The algorithm should update the facial areas that have been occluded in the first frame. The simplest idea is to refresh the subject's texture for the frames with minimum fitting ABS error. Also, since the first texture is already illuminated by an unknown light configuration it is imperative that the fitting algorithm include an illumination model.

The quality of the synthesized facial images can be evaluated using a simple quality measure, namely the peak-signal-to-noise-ratio (PSNR). PSNR compares the reconstructed synthetic image with the original one, and is defined as:

$$PSNR = 10\log\frac{255^2}{\frac{1}{N}\sum_{i-0}^{N-1}(I_{orig,i} - I_{synth,i})^2} \tag{6.1}$$

Where $I_{orig,i}$ is the intensity of pixel i in the original image with values from ranging from 0 to 255, and $I_{synth,i}$ is the corresponding value in the synthetic image, and N is the pixel count of the facial area. PSNR should evaluate the encoded object, in this case, the facial area only without the background. Without representing an absolute visual quality perceived by the human eye, PSNR allows comparing results produced by the same class of codecs [Eis00]. Eisert's [Eis00] MBVC achieved an average of 31.6 dB PSNR at bit-rates as low as 0.6 kps. Additional encoding of illumination and texture update parameters will lead to a minor increase in bit-rate per frame and a significant increase in reconstruction quality.

Appendix

A.1. ANTHROPOMETRIC LANDMARKS AND MEASUREMENTS

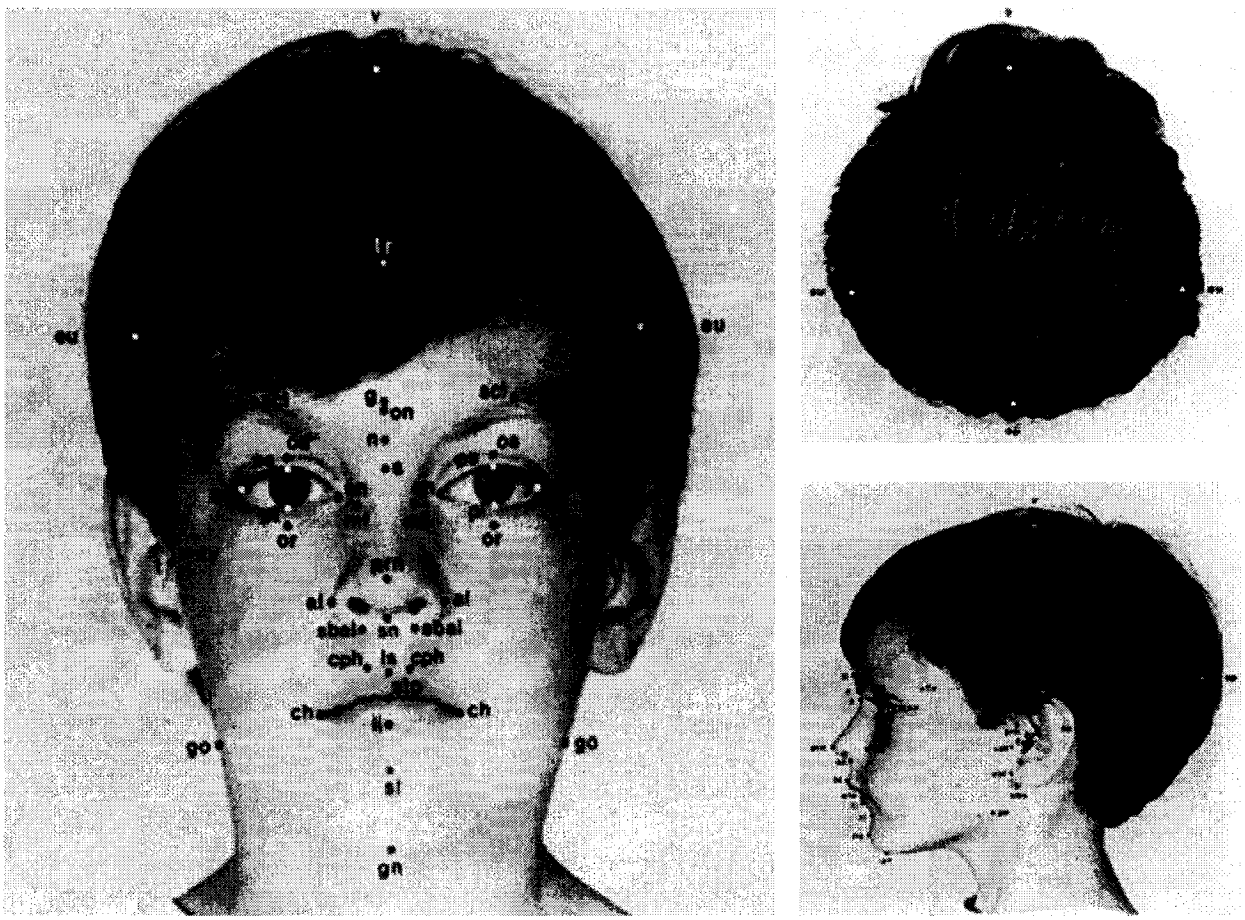


Fig. a.1: Craniofacial landmarks [Kol96]

No.	Landmark	Region	Definition
1	Euryon (eu)	Head	The most lateral point on the head
2	Frontotemporal (ft)	Head	The most medial point on the temporal crest of the frontal bone
3	Frontozygomaticus (fz)	Head	The most lateral point on the frontozygomatic suture
4	Glabella (g) or nasal eminence	Head	The most prominent point in the median sagital plane between the supra-orbital ridges
5	Ophyron (on) or point	Head	The point at the mid-plane of a line tangent to the

	intersoucilier		upper limits of the eyebrows (sci-sci)
6	Opisthorcranion (op) or occipital point	Head	The most prominent posterior point of the occiput
7	Porion (po)	Head	The most superior point on the upper margin of the external auditory meatus when the head is in the Frankfort horizontal plane
8	Tragion (t)	Head	Located at the notch above the tragus of the ear, the cartilaginous projection in the front of the external auditory canal, where the upper edge of the cartilage disappears into the skin of the face
9	Trichion (tr)	Head	Midpoint of the hairline
10	Vertex (v)	Head	The highest point of the head with the subject in the Frankfurt horizontal plane
11	Condylion laterale (cdl)	Face	The most lateral point on the mandibular condyle
12	Gnathion (gn) or Menton	Face	The lowest point in the midline on the lower border of the chin
13	Gonion (go)	Face	The most lateral point at the angle of the mandible
14	Nasion (n)	Face	The midpoint of the nasofrontal suture
15	Pogonion (pg)	Face	The most anterior point in the middle of the soft tissue chin
16	Sublabial (sl)	Face	The midpoint of the Labiomental sulcus
17	Subnasal (sn)	Face	The junction between the lower border of the nasal septum, the partition that divides the nostrils, and the cutaneous portion of the upper lip in the midline
18	Stomion (sto) or point buccal	Face	The mid point of the labial fissure when the lips are closed naturally
19	Zygion (zy)	Face	The most lateral point on the zygomatic arch

Table a.1: Craniofacial landmarks

A.2. EXTENDED KALMAN FILTER

The Linear Kalman Filter (LKF) addresses the general problem of estimating the state of a discrete-time controlled process governed by a *linear* stochastic difference-equation. When the process to be estimated is non-linear, one has to use a non-linear estimation technique (EKF) to solve the SFM problem. Essentially the EFK is applied to nonlinear systems with additive white noise. The EFK continually updates a linearization around the previous state estimate, starting with an initial guess. The EFK linearizes about the current mean and covariance values. The degree of accuracy of this linearization depends on the initial guess and the evolution of disturbances. The formalism assumes that the system-noise and measurement-noise processes are uncorrelated and both are Gaussian white-noise sequences. Similarly to a Taylor series, the filter linearizes the estimation around the current estimate using the partial derivatives of the process and measurement functions to compute estimates even in the case of non-linear relationships.

A non-linear dynamic system can be represented by the following equations:

$$s(k+1) = f(s(k), w(k)) \tag{a.1}$$

$$m(k) = h(s(k), v(k)) \tag{a.2}$$

where:

- $w(k)$ and $v(k)$ are the random variables that represent the process and measurement noise. We assumed them to be independent of each other, white, and with normal probability distributions.

  $$p(w) \sim N(0, Q) \tag{a.3}$$

  $$p(v) \sim N(0, R) \tag{a.4}$$

- the *non-linear* function $f(\bullet)$ in the difference equation (a.1) relates the state at time step k to the state at step $k+1$. It includes as parameter the zero-mean process noise $w(k)$
- the *non-linear* function $h(\bullet)$ in the measurement equation (a.2) relates the state $s(k)$ to the measurement $m(k)$.

After linearization, the state and measurement equations (a.1) and (a.2) can be expressed as [Wel98]:

$$s(k+1) = As(k) + \xi(k) \tag{a.5}$$

$$m(k) = Hs(k) + \eta(k) \tag{a.6}$$

where:

- A is the Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to s
- H is the Jacobian matrix of partial derivatives of $h(\bullet)$ with respect to s
- $\xi(k)$ and $\eta(k)$ are independent random variables having zero mean and covariance matrices: WQW^T and VRV^T, with Q and R from (a.3) and (a.4). Here, W is the Jacobian matrix of partial derivatives of $f(\bullet)$ with respect to $w(k)$, and V is the Jacobian matrix of partial derivatives of $h(\bullet)$ with respect to $v(k)$.

FILTERING AND PREDICTION

The "filtering and prediction" relates to the state estimate, where $\hat{s}(k \mid k)$ is the state estimate after a measurement ("filtered estimate") and $\hat{s}(k \mid k-1)$ is the state estimate after a time update ("predicted estimate"). The EKF recursively estimates the state vector at each frame using the measurement vector $m(k)$, the state prediction $\hat{s}(k \mid k-1)$ and the state prediction error covariance matrix $P(k \mid k\text{-}1)$. $H(k)$ is updated at each iteration of the EKF loop in order to reflect the nonlinear mapping between the state parameters and measurements. At the time step k the computational EKF cycle proceeds as shown in Fig. a.3:

Step 1. **Input** a new measurement vector $m(k)$.

Step 2. **Update** the measurement-system Jacobian matrix:

$$H(k) = \left(\frac{\partial h(k)}{\partial s} \right)_{s=\hat{s}(k|k-1)}$$

Step 3. **Compute** the system error covariance matrix:

$$P(k \mid k-1) = A(k)P(k-1 \mid k-1)A^T(k) + W(k-1)Q(k-1)W^T(k-1)$$

Step 4. **Compute** the gain matrix:

$$K(k) = P(k \mid k-1)H^T(k)[H(k)P(k \mid k-1)H^T(k) + V(k)R(k)V^T(k)]^{-1}$$

Step 5. **Update** the system error covariance matrix:

$$P(k \mid k) = (I - K(k)H(k))P(k \mid k-1)$$

Step 6. **Predict** the state vector:

$$\hat{s}(k \mid k-1) = f(\hat{s}(k-1 \mid k-1),0)$$

Step 7. **Update** the state vector:

$$\hat{s}(k \mid k) = \hat{s}(k \mid k-1) + K(k)[m(k) - h(\hat{s}(k \mid k-1),0)]$$

Step 8. **Store** for the next cycle the updated state vector and system error covariance matrix:

$$\hat{s}(k \mid k) \xrightarrow{step+} \hat{s}(k-1 \mid k-1)$$

$$P(k \mid k) \xrightarrow{step+} P(k-1 \mid k-1)$$

Fig. a.3: The EKF algorithm steps.

EFK equations suggest that the filter behavior agree with our human intuition. The gain matrix can be expressed [Zha92] in the form:

$$K(k) = P(k \mid k)H^T(k)R^{-1}(k), \qquad \text{(a.7)}$$

Hence, the Kalman gain is "proportional" to the uncertainty in the estimate and "inversely proportional" to the measurement uncertainty. Having defined the residual as $r = m(k) - h(\hat{s}(k \mid k-1),0)$, the expression in (a.7) reflects two scenarios:

- very uncertain *measurement* and relatively precise *state estimate.* In this case the residual r is generated by the noise and the state estimate needs little change.
- relatively precise *measurement* and very uncertain *state estimate.* In this case the residual r mirrors the state estimate errors and the state estimate needs a strong correction.

Inverting $P(k|k)$ and replacing $K(k)$ in (a.7), we obtain the covariance matrix equation:

$$P^{-1}(k \mid k) = P^{-1}(k \mid k-1) + H^T(k)R^{-1}(k)H(k), \qquad \text{(a.8)}$$

We notice the "proportionality" between P and R covariance matrices. It also can be seen that a very precise measurement ($R(k)$ is small) will considerably reduce the estimation error by decreasing the estimation variance.

B. REFERENCES

[Ahl98] J. Ahlberg and H. Li, *Representing and compressing MPEG-4 facial animation parameters using facial action basis functions,* Technical Report, LiTH-ISY-R-2010, ISSN 1400-3902, Linkoping University, 1998.

[Ahl99] J. Ahlberg, *Extraction and Coding of Face Model Parameters*, Licentiate Thesis No. 747, Dept. of Electrical Engineering, Linköpings Universitet, Sweden, 1999.

[Ahl03] J. Ahlberg and R. Forchheimer, *Face tracking for model-based coding and face animation*, Int. Journal of Imaging Systems and Technology, vol. 13, pp. 8-22, 2003.

[Aiz87] K. Aizawa, H. Harashima, and T. Saito, *A model-based image coding system construction of a 3-D model of a person's face*, Proc. of the Int. Picture Coding Symposium, paper 3.11, Stockholm, Sweden, 1987.

[Aiz89] K. Aizawa, H. Harashima, and T. Saito, *Model-based analysis synthesis image coding for a person's face,* Image Communication, vol. 1, no. 2, pp.139-152, 1989.

[Aiz95] K. Aizawa, T.S. Huang, *Model Based Image Coding: Advanced Video Coding Techniques for very Low Bit-Rate Applications,* Proc. IEEE, vol. 3, no. 2, pp. 259-271, Feb. 1995.

[Alt92] T.D. Alter, *3d pose from 3 corresponding points under weak-perspective projection,* Technical Report 1378, MIT Artificial Intelligence Laboratory, 1992.

[Aza93] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, *Visually controlled graphics,* Trans.Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 730-742, 1993.

[Aza95] A. Azarbayejani and A. Pentland, *Recursive estimation of motion, structure, and focal length*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 6, June, pp. 562-575, 1995.

[Bak01] S.Baker and I.Matthews, *Equivalence and efficiency of image alignment algorithms*, Computer Vision and Pattern Recognition Conf. 2001, vol. 1, pp. 1090–1097, 2001.

[Bar99] M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, *Measuring facial expressions by computer image analysis*, Psychophysiology, vol. 36, pp. 253–263, 1999.

[Bar04] M. S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, and J. R. Movellan, *Machine learning methods for fully automatic recognition of facial expressions and facial actions*, Proc. of IEEE Int. Conf. Systems, Man, and Cybernetics, pp. 592–597, 2004.

[Bas96] S. Basu, I. Essa, and A. Pentland, *Motion regularization for model-based head tracking*, Int. Conf. on Pattern Recognition, Vienna, Austria, 1996.

[Bass79] J. N. Bassili, *Emotion Recognition: The Role of Facial Movement and the Relative Importance of Upper and Lower Areas of the Face*, Journal of Personality and Social Psychology, vol. 37, pp. 2049-2058, 1979.

[Bla99] V. Blanz and T. Vetter, *A morphable model for the synthesis of 3D faces*, Computer Graphics, Annual Conf. Series (SIGGRAPH), pp.187-194, 1999.

[Bon01] M. D. Bondy, E. M. Petriu, M. D. Cordea, N. D. Georganas, D. C. Petriu, and T. E. Whalen, *Model-based Face and Lip Animation for Interactive Virtual Reality Applications*, Proc. of ACM Multimedia 2001, pp. 559-563, Ottawa, ON, Canada, Sept. 2001.

[Bla97] M.J. Black and Y. Yacoob, *Recognizing Facial Expressions in Image Sequences Using Local Parameterized Models of Image Motion*, Int. Journal of Computer Vision, vol. 25, no. 1, pp. 23-48, 1997.

[Bro86] T. J. Broida and R. Chellappa, *Estimation of object motion parameters from noisy images*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, no. 1, pp. 90-99, Jan. 1986.

[Bux01] B.F. Buxton T.J. Hutton and P. Hammond, *Dense surface point distribution models of the human face*, Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, pp. 153-160, Kauai, Hawaii, 2001.

[Cas98] M. La Cascia, J. Isidoro, S. Sclaroff, *Head tracking via robust registration in texture map images*, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 508-514, June. 1998.

[Cha94] A. Chakraborty, L. H. Staib, and J. S. Duncan, *Deformable boundary finding influenced by region homogeneity,* Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, pp. 624-627, Seattle, WA, June 1994.

[Chr04] C. Christoudias, M., L. Morency, and T. Darrell, *Light Field Appearance Mainfolds*, Proc. of the European Conf. on Computer Vision (ECCV '04), Prague, Czech Republic, pp. 481-493, May 2004.

[Coh91] L. Cohen, *Note on active contour models and balloons,* CVGIP: Image Understanding, vol. 53, no. 2, pp. 211-218, March 1991.

[Coh03] I. Cohen, N. Sebe, A. Garg, L. Chen, and T. Huang, *Facial expression recognition from video sequences: temporal and static modeling*, Computer Vision and Image Understanding, vol. 91, no.1 , pp. 160–187, 2003.

[Coh98] J.F. Cohn, A.J. Zlochower, J.J. Lien, and T. Kanade, *Feature-Point Tracking by Optical Flow Discriminates Subtle Differences in Facial Expression*, Proc. of Int. Conf. Automatic Face and Gesture Recognition, pp. 396-401, 1998.

[Coh99] J. F. Cohn, A. J. Zlochower, J. Lien, and T. Kanade, *Automated face analysis by feature point tracking has high concurrent validity with manual faces coding*, Psychophysiology, vol. 36, pp. 35–43, 1999.

[Coo95] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham, *Active shape models - their training and application*, Computer Vision and Image Understanding, no. 61, vol. 1, pp. 38-59, Jan. 1995.

[Coo98] T.F. Cootes, G.J. Edwards, and C.J. Taylor, *Active Appearance Models*, Proc. of the 5th European Conf. on Computer Vision (ECCV), pp. 484-498, Freiburg, Germany, June 1998.

[Coo98a] T. F. Cootes, G. J. Edwards, and C. J. Taylor, *A comparative evaluation of active appearance model algorithms*, British Machine Vision Conf., vol. 2, pp. 680–689, Southampton, UK, BMVA Press, Sept. 1998.

[Coo99] T.F Cootes and C.J. Taylor, *Statistical Models of Appearance for Computer Vision*, Tech. Report, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, University of Manchester, Sept. 1999.

[Coo99a] T.F. Cootes, G. J. Edwards and C. J. Taylor, *Comparing Active Shape Models with Active Appearance Models*, Proc. of British Machine Vision Conf., vol. 1, pp. 173-182, 1999.

[Coo00] T. F. Cootes, K. N. Walker, and C. J. Taylor, *View-based active appearance models,* Proc. 4th Int. Conf. on Automatic Face and Gesture Recognition 2000, pp. 227-232, Grenoble, France, 2000.

[Coo01] T. F. Cootes and C. J. Taylor, *Statistical models of appearance for medical image analysis and computer vision*, Proc. of SPIE Medical Imaging, vol. 3, pp. 138-147, 2001.

[Coo02] T.F. Cootes and P.Kittipanya-ngam, *Comparing Variations on the Active Appearance Model Algorithm*, Proc. BMVC, vol.2, pp.837-846, 2002.

[Cor98] M. D. Cordea, E. M. Petriu, D. C. Petriu, *Object-Oriented Face Animation For Model-Based Video Compression Applications*, ICT98- Porto Caras, Greece, June 1998.

[Cor01] M.D. Cordea, E. M. Petriu, N.D. Georganas, D.C. Petriu, and T.E. Whalen, *Real-Time 2½D Head Pose Recovery for Model-Based Video-Coding*, IEEE Trans. Instrum. Meas., vol. 50, no. 4, pp.1007-1013, 2001.

[Cor02] M.D. Cordea, D.C. Petriu, E.M. Petriu, N.D. Georganas, and T.E.Whalen, *3-D Head Pose Recovery for Interactive Virtual Reality Avatars*, IEEE Trans. Instrum. Meas., vol. 51, no. 4, pp. 640-644, 2002.

[Cor04] M.D. Cordea, E.M. Petriu, *A 3D-Anthropometric-Muscle-Based Active Appearance Model*, Proc. of VECIMS'04, IEEE Int. Conf. on Virtual Environments, Human-Computer Interfaces and Measurement Systems, pp. 88-93, Boston, MA, July 2004.

[Cor06] M.D. Cordea, E.M. Petriu, *A 3D-Anthropometric-Muscle-Based Active Appearance Model*, IEEE Trans. Instrum. Meas., vol. 55, no. 1, pp.91 - 98, 2006.
[Dar896] C. R. Darwin, *The expression of emotions in man and animals*, New York: Appleton, 1896.

[Dav02] R. H. Davies, *Learning Shape: Optimal Models for Analysing Natural Variability,* PhD. Thesis, University of Manchester, 2002.

[Dav03] R. H. Davies, C.J. Twining, T.F. Cootes, J.C. Waterton and C.J. Taylor, *A Minimum Description Length Approach to Statistical Shape Modelling*, IEEE Trans. Medical Imaging, vol.21, no.5, pp. 525-537, 2002.

[DeC96] D. DeCarlo and D. Metaxas, *The Integration of Optical Flow and Deformable Models with Applications to Human Face Shape and Motion Estimation*, Computer Vision Pattern Recognition, pp. 231, June 1996.

[DeC98] D. DeCarlo, D. Metaxas, and M. Stone, *An Anthropometric Face Model using Variational Techniques*, Department of Computer and Information Science, University of Pennsylvania, Proc. of SIGGRAPH'98, pp. 67-74, 1998.

[Dav97] M. Davis and M. Tuceryan, *Coding of Facial Image Sequences by Model-Based Optical Flow*, Int. Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI'97), Rodos-Palace, Rhodes, Greece, Sept. 5-9, 1997.

[Dev01] V. E. Devin and D. C. Hogg, *Reactive Memories: An interactive talking-head*, Research Report Series, School of Computing, University of Leeds, Report, Sept. 2001.

[Dor03] Fadi Dornaika and Jorgen Ahlberg, *Fast and reliable Active Appearance Model Search for 3D face tracking,* Proc. of Mirage, INRIA Rocquencourt, France, pp. 10-11, March 2003.

[Edw98] G.J. Edwards, T.F. Cootes, and C.J. Taylor, *Interpreting Face Images using Active Appearance Models*, Proc. of the 3rd IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG), pp. 300–305, Nara, Japan, April 1998.

[Edw98a] G. J. Edwards, C.J. Taylor, and T.F. Cootes, *Learning to identify and track faces in image sequences*, Int. Conf. on Automatic Face and Gesture Recognition, pp. 317-322, 1998.

[Edw98b] G.J. Edwards, T.F. Cootes, and C.J. Taylor, *Face Recognition Using Active Appearance Models*, Proc. of European Conf. Computer Vision, vol. 2, pp. 581-695, 1998.

[Eis97] P. Eisert and B. Girod, *Model-based 3D motion estimation with illumination compensation*, Proc. of Int. Conf. on Image Proc. and its Applications, vol. 1, pp. 194-198, 1997.

[Eis00] P. Eisert, *Very Low Bit-Rate Video Coding Using 3-D Models*, PhD thesis, University of Erlangen, Shaker Verlag, Aachen, Germany, 2000.

[Eis02] P. Eisert and B. Girod, *Model-based enhancement of lighting conditions in image sequences*, Proc. Visual Communication and Image Processing, San Jose, CA, Jan. 2002.

[Ekm71] P. Ekman, and W. Friesen, *Constants across Cultures in the Face and Emotion*, Journal of Personality and Social Psychology, vol. 17, no. 2, pp. 124-129, 1971.

[Ekm77] P. Ekman, and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of the Facial Movement*, Consulting Psychologists Press Palo Alto, 1977.

[Ekm86] P. Ekman, and W. Friesen, *Manual for the Facial Action Coding System*, Consulting Psychologists Press, Palo Alto, 1986.

[Ekm83] P. Ekman, W.V. Friesen, *EMFACS-7: Emotional Facial Action Coding System*, Unpublished manuscript, University of California at San Francisco, 1983.

[Eps95] R. Epstein, P. Hallinan, and A. Yuille, *5 plus or minus 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models*, Proc. of IEEE Work. Physics-based Modeling in Comp. Vision, 1995.

[Ers03] M. B. Stegmann, B. K. Ersbøll, and R. Larsen, *FAME - a flexible appearance modeling environment*, IEEE Trans. on Medical Imaging, vol. 22, pp. 1319-1331, May 2003.

[Ess97] I. Essa and A. Pentland, *Coding, Analysis Interpretation, Recognition of Facial Expressions*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 757-763, July 1997.

[Far94] L. Farkas, *Anthropometry of the Head and Face*, Raven Press, 1994.

[Fas03] B. Fasel and J. Luettin, *Automatic facial expression analysis: A survey*, Pattern Recognition, vol. 36, no.1, pp. 259-275, 2003.

[Fau92] O. Faugeras, *What can be seen in three dimensions from an uncalibrated stereo rig*, Proc., 2nd European Conf. on Computer Vision, pp. 563-578, Springer-Verlag, Santa Margherita Ligure, Italy, 1992.

[FGnet] FGnet - IST-2000-26434, Face and Gesture Recognition Working Group, [Online]: http://www.mmk.ei.tum.de/~waf/fgnet/feedtum.html

[Fin02] G. D. Finlayson, S. D. Hordley, M. S. and Drew, *Removing shadows from Images using retinex*, 10th Color Imaging Conf.: Color Science and Engineering Systems, Technologies, Applications, Scottsdale, Arizona, USA, pp. 73-79, 2002.

[Flo00] *The Flock of Birds*, Ascension Technology Corporation, P.O. Box 527, Burlington, VT 05402.

[For83] R. Forchheimer and O. Fahlander, *Low bit-rate coding through animation*, Proc. of the Picture Coding Symposium, Davis, CA, USA, pp. 113-114, March 1983.

[Fra83] J Frankle and J. McCann, *Method and apparatus for lightness imaging*, US Patent No. 4,384,336, May 1983.

[Fun99] B. Funt and K Barnard, *Investigation into Multi-Scale Retinex (MSR),* Color Imaging: Vision and Technology, New York: Wiley, pp. 9-17, 1999.

[Gen82] D.B. Gennery, *Tracking known 3-dimensional objects,* Proc. AAAI 2nd National Conf. on Artificial Intelligence, pp. 13–17, Pittsburg, PA, 1982.

[Gen92] D.B. Gennery, *Visual tracking of known 3-dimensional object,* Int. Journal of Computer Vision, vol. 7, no. 3, pp. 243–270, 1992.

[Geo99] N.D. Georganas, E. Petriu, M. Cordea, D. Ionescu, *Distributed Virtual Environments for Training and Telecollaboration*, Proc. IMTC/98, IEEE Instrum. Meas. Technol. Conf., pp. 1847-1850, Venice, Italy, May 1999.

[Gir94] B. Girod, *Image sequence coding using 3D scene models*, Symposium on Visual Communications and Image Processing, Sept. 1994.

[Gow75] J. C. Gower, *Generalized Procrustes analysis*, Psychometrika, 40, pp. 33-50, 1975.

[Gro04] R. Gross, I. Matthews, and S. Baker, *Appearance-based face recognition and light fields*, IEEE Pattern Analysis and Machine Intelligence, vol. 26, no. 4, pp. 449-465, April 2004.

[Hac03] Hack, C. and Taylor, CJ, *Modelling talking head behavior*, Proc. of British Machine Vision Conf., vol. 1, pp. 33-42, 2003.

[Hag96] G. D. Hager and P. N. Buelhumeur, *Real-time tracking of image regions with changes in geometry and illumination*, IEEE Conf. in Computer Vision Pattern Recognition, pp. 403–410, 1996.

[Han02] D.W. Hansen, M. Nielsen, J. PP. Hansen, A. S. Johansen and M. B. Stegmann, *Tracking eyes using shape and appearance*, IAPR Workshop on Machine Vision Applications, pp. 201-204, 2002.

[Har94] R. Hartley, *Lines and points in three views - an integrated approach*, Proc. ARPA IU Workshop, ARPA94, pp. 1009-1016, 1994.

[Hea95] T. Heap, D. C. Hogg, *Automated Pivot Location for the Cartesian-Polar Hybrid Point Distribution Model*, British Machine Vision Conf., MVC'95, University of Birmingham, UK, pp. 97-106, 1995.

[Hor74] B.K.P. Horn, *Determining lightness from an image*, Computer Vision, Graphics, and Image Processing, no. 3, pp. 277-299, 1974.

[Hor86] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT, 1986.

[Hon98] H. Hong, H. Neven, and C. von der Malsburg, *Online Facial Expression Recognition Based on Personalized Galleries*, Proc. Int. Conf. Automatic Face and Gesture Recognition, pp. 354-359, 1998.

[Hou01] X. Hou, S. Li, H. Zhang, and Q. Cheng, *Direct appearance models*, Computer Vision and Pattern Recognition Conf. 2001, vol. 1, pp. 828–833, 2001.

[Hua97] C.L. Huang and Y.M. Huang, *Facial Expression Recognition Using Model-Based Feature Extraction and Action Parameters Classification*, J. Visual Comm. and Image Representation, vol. 8, no. 3, pp. 278-290, 1997.

[Hua01] T. S. Huang and H. Tao, *Visual Face Tracking and its Application to 3D Model-based Video Coding*, Picture Coding Symposium, pp. 57-60, 2001.

[Hwa89] V. S. S. Hwang, *Tracking feature points in time-varying images using an opportunistic selection approach*, Pattern Recognition, vol. 22, no. 3, pp. 247–256, 1989.

[ISO98] ISO/IEC Standard 14496-2, *Information Technology - Coding of audio-visual objects - Part 2: Video*, Int. Standard, First edition, Oct. 1998.

[Jai96] A. K. Jain, Y. Zhong, S. Lakshmanan, *Object matching using deformable templates,* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 18, no. 3, pp. 38-59, 1996.

[Jeb97] T. Jebara, A. Pentland, *Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces*, Proc. of Computer Vision and Pattern Recognition, 1997.

[Kan02] H. Kang, Timothy F. Cootes, Christopher J. Taylor, *A Comparison of Face Verification Algorithms using Appearance Models*, British Machine Vision Conf., 2002.

[Kan00] T. Kanade, J.F. Cohn, and Y. Tian, *Comprehesive database for facial expression analysis*, Proc. of 4rd Int. Conf. Automatic Face and Gesture Recognition, pp. 46–53, 2000.

[Kas87] M. Kass, A.Witkin, and D. Terzopoulos, *Snakes: Active contour models,* 1st Interntional Conf. on Computer Vision, pp. 259–268, London, June 1987.

[Koch96] R. M. Koch, M. H. Gross, F. R. Carls, D. F. von Büren, Y. I. H. Parish, *Simulating Facial Surgery Using Finite Element Models*, Proc. Computer Graphics, Annual Conf. Series, ACM SIGGRAPH, pp. 421-428, 1996.

[Kol96] J. C. Kolar, and E. M. Salter, *Craniofacial anthropometry practical measurements of the head and face for clinical, surgical and research use*, Charles Thomas publisher ltd., USA, 1996.

[Lan71] E. Land and J. McCann, *Lightness and Retinex Theory*, Journal of the Optical Society of America, no. 61, 1971.

[Lan86] E. Land, *Recent advances in Retinex theory*, Vision Res., no. 26, pp. 7-21, 1986.

[Lan97] A. Lanitis, C. J. Taylor, and T. F. Cootes, *Automatic interpretation and coding of face images using flexible models,* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 742–756, 1997.

[Lan99] A. Lantis, C. Taylor and T. Cootes, *Modelling the Process of Ageing in Face Images*, Proc. of 7th IEEE Int. Conf. on Computer Vision, vol. 1, pp. 131-136, 1999.

[Ley93] F. Leymarie, and M. Levine, *Tracking deformable objects in the plane using an active contour model,* IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 617-634; June 1993.

[Lie98] J. J. Lien, T. Kanade, J. F. Cohn, and C. C. Li, *Automated facial expression recognition based on FACS action units*, Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Rec. (FG'98), Nara, Japan, 1998.

[Lin04] Y. Lin, L. Liu, S. Fuh, *Fast object detection with occlusions*, Proc. 8th European Conf. in Computer Vision, vol. 1, pp. 402-413, 2004.

[LiH93] H. Li, P. Roivanen, and R. Forchheimer, *3-D Motion Estimation in Model-Based Facial Image Coding*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no 6, pp. 545-556, 1993.

[Low87] D. G. Lowe, *Three-dimensional object recognition from single two-dimensional images,* Artificial Intelligence, vol. 31, no. 3, pp. 355-395, March 1987.

[Lyo99] M.J. Lyons, J. Budynek, and S. Akamatsu, *Automatic Classification of Single Facial Images*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 12, pp. 1,357-1,362, 1999.

[Mar00] D. Marini and A. Rizzi, *A computational approach to colour adaptation effects*, Image and Vision Computing, no. 18, pp. 1005-1014, 2000.

[Mat89] L. Matthies, R. Szelisky, and T. Kanade, *Kalman filter-based algorithms for estimating depth from image sequences,* Int. Journal of Computer Vision, 1989.

[Mat99] Y. Matsumoto, A. Zelinsky, *Real-time Face Tracking System for Human-Robot Interaction*, Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC'99), pp. 830-835, Tokyo, Japan, Oct. 12-15, 1999.

[McC76] J. McCann, S. McKee, and T. Taylor, *Quantitative Studies in Retinex Theory, A Comparison between Theoretical Predictions and Observer Responses to the Color Mondrian Experiments*, Vision Research, no. 16, pp. 445-458, 1976.

[McI96] T. McInerney and D. Terzopoulos, *Deformable models in medical image analysis: a survey*, Medical Image Analysis, vol. 1, no. 2, pp. 91-108, 1996.

[Mit02] S.C. Mitchell, J.G. Bosch, B.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 Trans. on Medical Imaging 21, pp. 1167-1178, 2002.

[Moo99] K. L. Moore, and A. F. Dalley, *Clinically oriented anatomy*, 4th edition, Lippincott Williams & Wilkins, Philadelphia 1999.

[Mor03] L. P. Morency, P. Sundberg, T. Darrel, *Pose Estimation using 3D View-Based Eigenspaces*, IEEE Int. Workshop on Analysis and Modeling of Faces and Gestures, pp. 45-52, Nice, 2003.

[MPG02] Int. Organization for Standardization: ISO/IEC JTC1/SC29/WG11, *Coding of Moving Pictures and Audio – MPEG4 Overview*, N4668, Mar. 2002.

[Mur95] H. Murase and S. Nayar, *Learning and recognition of 3d objects from appearance*, Int. Journal of Computer Vision, pp. 5-25, Jan. 1995.

[Oli98] Olivetti Research Laboratory (ORL) database, http://www.orl.co.uk/facedatabase.html

[Ope05] Open Source Computer Vision Library (OpenCV), at http://www.intel.com/research/mrl/research/opencv

[Pan00a] M. Pantic and L.J.M. Rothkrantz, *Expert System for Automatic Analysis of Facial Expression*, Image and Vision Computing Journal, vol. 18, no. 11, pp. 881-905, 2000.

[Pan00b] M. Pantic and L. J. M. Rothkrantz, *Automatic Analysis of Facial Expressions: The State of the Art*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 12, pp. 1424-1445, 2000.

[Pan05] M. Pantic, M. F. Valstar, R. Rademaker, and L. Maat, *Web-based database for facial expression analysis*, in Proc. IEEE Conf. Multimedia and Expo, pp.317–321, 2005, [Online] Available at: http://www.mmifacedb.com

[Par94] F. I. Parke, and K. Waters, *Computer Facial Animation*, AK Peters, ISBN 1-56881-014-8, 1994.

[Pea95] D. E. Pearson, *Development in Model-Based Video Coding*, Proc. of IEEE, vol. 83, pp. 892-906, Jun. 1995.

[Pen94] A. Pentland, B. Moghaddam, and T. Starner, *View-Based and Modular Eigenspaces for Face Recognition*, Proc. of Computer Vision and Pattern Recognition, pp. 84-91, Jun. 1994.

[Per02] F. Pereira and T. Ebrahimi, *The MPEG-4 Book*, Prenticle Hall PTR, 2002.

[Phi99] J. Phillips, H. Moon, S. Rizvi, and P. Rauss, *The FERET Evaluation Methodology for Face-Recognition Algorithms*, NIST Technical Report NISTIR 6264, 1999.

[Ram02] R. Ramamoorthi, *Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 10, Oct. 2002.

[Ris83] J. R. Rissanen, *A universal prior for integers and estimation by minimum description length*, Annals of Statistics, vol. 11, no. 2, pp. 416-431, 1983.

[Rom99] S. Romdhani, S. Gong, and A. Psarrou, *A multi-view non-linear active shape model using kernel pca*, 10th British Machine Vision Conf., vol. 2, pp. 483-492, Nottingham, UK, Sept. 1999.

[Rom03] S. Romdhani, T. Vetter, *Efficient, Robust and Accurate Fitting of a 3D Morphable Model*, Proc. Int. Conf. on Computer Vision, vol. 1, pp. 59-66, 2003.

[Ryd87] M. Rydfalk, *CANDIDE, a parameterized face*, Report no. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linköping University, Sweden, 1987.

[Sca98] S. Sclaroff and J. Isidoro, *Active Blobs*, Int. Conf. on Computer Vision, pp. 1146–1153, Mumbai, India, 1998.

[Sch97] B. Scholkopf, A. Smola, and K. Muller, *Kernel principal component analysis*, Artificial Neural Networks - CANN'97, 1997.

[Sha95] L. S. Shapiro, *Affine Analysis of Image Sequences*, Ph.D. dissertation, Sharp Lab. of Europe, Oxford, U.K., 1995.

[She92] C. Shekhar and R. Chellappa, *Passive ranging using a moving camera*, Journal of Robotics S. vol. 9, no.6, pp. 729-752 1992.

[Shi94] J. Shi and C. Tomasi, *Good features to track*, IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'94), pp. 593-600, Seattle, WA, June 1994.

[Sho85] K. Shoemake, *Animating Rotations with Quaternion Curves*, SIGGRAPH, pp. 245-254, 1985.

[Sho94] K. Shoemake, *Quaternions,* Department of Computer and Information Science University of Pennsylvania Philadelphia, PA, 1994

[Sko03] K. Skoglund, *Three-dimensional face modeling and analysis*, Master's thesis, IMM, Technical University of Denmark, Richard Petersens Plads, DK-2800 Kgs. Lyngby, Aug. 2003.

[Sou95] P. D. Sozou, T. F. Cootes, C. J. Taylor, and E. C. Di Mauro, *Non-linear point distribution modelling using a multi-layer perceptron,* 6th British Machine Vision Conf., pp. 107-116, Birmingham, 1995.

[Spo99] H.J.W. Spoelder, E. Petriu, T. Whalen, D.C. Petriu, M. D. Cordea, *Knowledge-Based Animation of Articulated Anthropomorphic Models for Virtual Reality Applications*, Proc. of IMTC/98, IEEE Instrum. Meas. Technol. Conf., pp. 690-695, Venice, Italy, May 1999.

[Sta95] J. Stauder, *Estimation of point light source parameters for object-based coding*, Signal Processing and Image Comm., vol. 7, no. 4, pp. 355-379, 1995.

[Ste00] M. B. Stegmann, R. Fisker, B. K. Ersbøll, H. H. Thodberg, and L. Hyldstrup, *Active appearance models: Theory and cases*, Proc. 9th Danish Conf. on Pattern Recognition and Image Analysis, Aalborg, Denmark, no. 1, pp. 49-57, 2000.

[Ste01] M. B. Stegmann, *Object tracking using active appearance models*, Proc. 10th Danish Conf. on Pattern Recognition and Image Analysis, no. 1, pp. 54-60, DIKU, 2001.

[Ste02] M. B. Stegmann, *Analysis and Segmentation of Face Images using Point Annotations and Linear Subspace Techniques*, Technical University of Denmark, IMM Technical Report IMM-REP-2002-22, Aug. 2002.

[Ste03] M. B. Stegmann and R. Larsen, *Multi-band modelling of appearance*, Image and Vision Computing, vol. 21, no. 1, pp. 61-67, 2003.

[Str99] J. Strom, T. Jebara, S. Basu, and A. Pentland, *Real Time Tracking and Modeling of Faces: An EKF-based Analysis by Synthesis Approach*, Proc. of Modelling People Workshop at ICCV'99, pp. 55, 1999.

[Str01] J. Strom, T. Jebara, and A. Pentland, *Model-Based Real-Time Face Tracking with Adaptive Texture Update*, Technical Report, LiTH-ISY-R-2342, Linköping University, Sweden, Mar. 30, 2001.

[Str02] J. Strom, *Model-Based Face Tracking and Coding,* Ph.D. Thesis, Dissertation No. 733, Department of Electrical Engineering, Linköping University, Sweden, February 2002.

[Sty03] A. Styner, T. Rajamani, P. Nolte, G. Zsemlye and, G. Szekely, J. Taylor, and H. Davies, *Evaluation of 3D Correspondence Methods for Model Building,* IPMI, 2003.

[Sze96] G. Szekely, A. Kelemen, C. Brechbuhler and G. Gerig, *Segmentation of 2-D and 3-D objects from MRI volume data using constrained elastic deformations of flexible Fourier surface models*, Medical Image Analysis, vol. 1, no. 1, 1996.

[Tay98] T. F. Cootes, G. Edwards, and C. J. Taylor, *A comparative evaluation of active appearance model algorithms*, Proc. 9th British Machine Vision Conf., vol. 2, pp. 680–689, Univ. of Southampton, 1998.

[Tek00] M. Tekalp, J. Ostermann, *Face and 2D Mesh Animation in MPEG-4*, Image Comm. Journal, Tutorial Issue on MPEG-4 Standard, Elsevier, 2000.

[Ter88] D. Terzopolous, A. Witkin, and M. Kass, *Constraints on deformable models: Recovering 3D shape and nonrigid motion,* Artificial Intelligence, vol. 36, pp. 91-123, 1988.

[Ter91] D. Terzopoulos and D. Metaxas, *Dynamic 3d models with local and global deformations: Deformable superquadrics,* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, no. 7, pp. 703–714, 1991.

[Ter91a] D. Terzopoulos, and M. Vasilescu, *Sampling and Reconstruction with Adaptive Meshes*, Proc. Conf. on Computer Vision and Pattern Recognition, Maui, HI, pp. 70-75, June 1991.

[Tia01] Y. Tian, T. Kanade, and J. Cohn, *Recognizing action units for facial expression analysis*, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 97-115, Feb. 2001.

[Tia03] Y. Tian, T. Kanade, and J. Cohn, *Handbook of face recognition*, chapter Facial expression analysis, Springer-Verlag, Heidelberg, 2003.

[Tur91] M. Turk and A. Pentland, *Eigenfaces for recognition*, Journal of Cognitive Neuroscience, vol. 1, no. 3, pp. 71–86, 1991.

[Vio01] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, Computer Vision and Pattern Recognition, 2001.

[Wan98] M. Wang, Y. Iwai, and M. Yachida, *Expression Recognition from Time-Sequential Facial Images by Use of Expression Change Model*, Proc. of Int. Conf. Automatic Face and Gesture Recognition, pp. 324-329, 1998.

[Wat87] Waters, K, *A Muscle Model for Animating 3D Facial Expressions*, Computer Graphics, vol. 21, no. 4, July 1987.

[Wel91] B. Welsh, *Model-Based Coding of Images*, PhD dissertation, British Telecom Research Lab, 1991.

[Wel98] G. Welch, G. Bishop, *An Introduction to the Kalman Filter,* Tech. Report 95-041, University of North Carolina at Chapel Hill, Oct. 1998.

[Wen03] Z. Wen, T. S. Huang, *Capturing Subtle Facial Motions in 3D Face Tracking*, Proc. Int. Conf. on Computer Vision, pp. 1343-1350, 2003.

[Wis95] L. Wiskott, *Labelled Graphs and Dynamic Link Matching for Face Recognition and Scene Analysis*, Reihe Physik, vol. 53, Frankfurt a.m. Main, 1995.

[Xia04] J. Xiao, S. Baker, I. Mattews and T. Kanade, *Real-Time Combined 2D+3D Active Appearance Models*, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, June, 2004.

[Yon97] M. Yoneyama, Y. Iwano, A. Ohtake, and K. Shirai, *Facial Expressions Recognition Using Discrete Hopfield Neural Networks*, Proc. Int. Conf. Information Processing, vol. 3, pp. 117-120, 1997.

[Yul92] A. Yuille, P. Hallinan, and D. Cohen, *Feature Extraction from Faces using Deformable Templates,* In Int. Journal of Computer Vision, vol. 8, no. 2, pp. 99-111, 1992.

[Zha92] Z. Zhang and O. Faugeras, *3D Dynamic Scene Analysis*, Springer Series in Information Sciences, 1992.

[Zha98] Z. Zhang, M. Lyons, M. Schuster, and S. Akamatsu, *Comparison between Geometry-Based and Gabor Wavelets-Based Facial Expression Recognition Using Multi-Layer Perceptron*, Proc. Int. Conf. Automatic Face and Gesture Recognition, pp. 454-459, 1998.