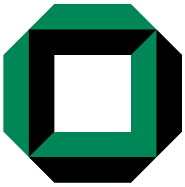


UNIVERSITÄT KARLSRUHE (TH)
FAKULTÄT FÜR INFORMATIK
INTERACTIVE SYSTEMS LABS
Prof. Dr. A. Waibel

DIPLOMA THESIS

Face Registration with Active Appearance Models for Local Appearance-based Face Recognition

SUBMITTED BY

Hua Gao

JUNE 2008

ADVISORS

M.Sc. Hazım Kemal Ekenel
Dr.-Ing. Rainer Stiefelhagen
Prof. Dr. Alex Waibel

Interactive Systems Labs
Institut für Theoretische Informatik
Universität Karlsruhe (TH)
Title: Face Registration with Active Appearance Models for Local Appearance-based Face Recognition
Author: Hua Gao

Hua Gao
Werderstr. 76
76137 Karlsruhe, Germany
email: hua.gao@ira.uka.de

Statement of authorship

I hereby declare that this thesis is my own original work which I created without illegitimate help by others, that I have not used any other sources or resources than the ones indicated and that due acknowledgement is given where reference is made to the work of others.

Karlsruhe, 30. June 2008

.....
(Hua Gao)

Abstract

Face recognition has received increasing attention from diverse research communities and the market over the past years. Various techniques have been intensively investigated aiming at high recognition accuracy and robustness against numerous facial appearance variations. Application areas of face recognition have also been expanded and more robust systems are required as the application scenarios become more unconstrained.

In this work, variation in facial appearance caused by 3D head pose was considered. The problem is also known as the face registration problem, which is an important factor for face recognition as demonstrated in many previous studies. The registration approach studied in this thesis is able to normalize the head pose in some degree of rotation in depth and align the face into a common coordinate framework. Moreover, the quality of face registration is assessed so that only successfully registered face images are used for recognition.

The developed face registration approach is based on active appearance model (AAM) fitting. A generic model was built in which both shape and appearance variations were modeled. After fitting the model on an input image, the pose of the input face was normalized and a frontal view of the input face was synthesized. To mitigate the influence of poor illumination, a modified histogram fitting approach was employed. Progressive model fitting was also investigated for a more robust estimate of model initialization. Face recognition was based on the fitted and pose normalized face images using our local appearance-based approach.

Three experiments were conducted to evaluate the AAM-based face registration approach. The first experiment was designed to evaluate the pose correction based on AAM fitting in still images. The results showed a significant improvement in face recognition performance compared to the previous affine-based registration approach, which again demonstrated the importance of pose correction for face recognition. We also compared our local appearance-based face recognition approach with two well known holistic approaches. The local appearance-based approach significantly outperformed the holistic approaches and it was more robust against the error introduced by AAM fitting and face synthesis. The second experiment evaluated the eye localization with AAM fitting. Face tracking with AAM fitting was also evaluated on a video database for open set face recognition. A modified distance from feature space metric was employed to assess the quality of fitting on a single frame. Open set face recog-

nition was performed on the successfully registered frames. The experimental results showed that both pose correction and registration quality assessment improved the recognition performance.

Kurzzusammenfassung

Gesichtserkennung wurde in den letzten Jahren sowohl von Seiten diverser Forschungsgemeinden, als auch vom Markt, große Aufmerksamkeit zuteil. Viele Methoden wurden intensiv untersucht mit dem Ziel, hohe Erkennungsleistung und Robustheit gegen Variationen in der Ansicht der Gesichter zu erreichen. Zudem wurden neue Anwendungsgebiete für Gesichtserkennung erschlossen, die robustere Systeme erfordern, da die Anwendungsszenarien immer weniger Einschränkungen haben.

In dieser Arbeit werden Variationen der Gesichtsansicht, die durch 3D Kopfdrehungen verursacht werden, behandelt. Dieses Problem ist bekannt als Gesichtsregistrirungsproblem und ist ein sehr wichtiger Einfluss auf die Leistung von Gesichtserkennungssystemen, wie in vielen Studien gezeigt wurde. Der Registrierungsansatz, der in dieser Arbeit behandelt wird, kann bis zu einem gewissen Grad Kopfdrehungen normalisieren und das Gesicht in ein gemeinsames Koordinatensystem transformieren. Zudem wird die Qualität der Registrierung gemessen, so dass nur erfolgreich normalisierte Gesichter zur Gesichtserkennung verwendet werden können.

Die entwickelte Gesichtsregistrierungsmethode basiert auf dem "fitten", d.h. der Parameterschätzung mit "active appearance models" (AAMs). Ein generisches Modell wurde generiert, bei dem sowohl Form- als auch Textur-Variationen modelliert wurden. Nachdem das Modell an ein Eingabebild "gefittet" wurde, wird die Kopfdrehung normalisiert und eine Frontalansicht des Gesichtes synthetisiert. Um den Einfluss schlechter Beleuchtungsverhältnisse zu minimieren, wird ein modifizierter Histogrammspezifikationsalgorithmus verwendet. Progressives AAM fitting wurde ebenfalls untersucht, um eine robustere Initialisierung des Systems zu erreichen. Gesichtserkennung wurde mit den normalisierten Gesichtsbildern mit Hilfe unseres auf lokalen Ansichten basierenden Ansatzes durchgeführt.

Um die AAM-basierte Gesichtsregisitrierung zu evaluieren wurden drei Experimente durchgeführt. Da erste Experiment evaluiert die Qualität der Normalisierung der Kopfdrehung mit Hilfe von Einzelbildern. Die Resultate zeigen einen signifikanten Anstieg der Gesichtserkennungsleistung, verglichen mit dem vorherigen mit einer affinen Transformation arbeitenden Ansatz, was wiederum die Wichtigkeit der Normalisierung von Kopfdrehungen für die Gesichtserkennung zeigt. Der auf lokalen Ansichten basierende Gesichtserkennungsalgorithmus wurde auch mit zwei bekannten holistischen Ansätzen verglichen. Er zeigte

deutliche bessere Ergebnisse und war auch robuster gegen Fehler, die aus AAM fitting und Synthese der Frontalansicht hervorgehen können. Das zweite Experiment evaluiert die Lokalisation der Augen mit Hilfe von AAM fitting. Gesichtstracking mit AAM fitting wurde ebenfalls mit Hilfe einer Video-Datenbank für "open-set" Gesichtserkennung evaluiert. Eine modifizierte Distanz-zum-Merkmalsraum-Metrik wurde verwendet, um die Qualität des fittings bei einem Einzelbild zu messen. Auf den erfolgreich normalisierten Bildern wurde dann "open-set" Gesichtserkennung durchgeführt. Die Ergebnisse zeigen, dass sowohl die Normalisierung der Kopfdrehung, als auch die Messung der Qualität der Registrierung, die Erkennungsleistung erhöhen.

Contents

List of Figures

List of Tables

List of Abbreviations

3DMM	3D morphable model
AAM	Active appearance model
ASM	Active shape model
CCR	Correct classification rate
CRR	Correct rejection rate
DCT	Discrete cosine transform
DFFS	Distance from feature space
DIFS	Distance in feature space
DWT	Discrete wavelet transform
EBGM	Elastic bunch graph matching
EER	Equal error rate
EHMM	Embedded hidden Markov models
FCR	False classification rate
FEM	Finite element model
FRR	False rejection rate
FRS	Face recognition system
FRT	Face recognition technologies
HCI	Human computer interface
IC	Inverse compositional
LDA	Linear discriminant analysis
NN	Nearest neighbor

PCA	Principal component analysis
PO	Project-out
ROC	Receiver operating characteristic
SIC	Simultaneous inverse compositional
SICOV	Simultaneous inverse compositional for video
SVM	Support vector machine
TPS	Thin-plate splines

1 Introduction

In the past two decades, face recognition has received substantial attention from researchers in signal processing, pattern recognition, and computer vision communities [13, 69]. The motivation of the common interest is our remarkable ability to recognize faces and the importance of such ability for our daily life. Besides, face recognition technologies are increasingly required in a wide range of applications. Sample applications are surveillance, human computer interface (HCI), access control and content-based image/video management. Many commercial face recognition systems have been deployed, based on numerous research works during the past years. However, the performance of most systems is easily affected by many factors because the appearance of a human face has potentially very large variations due to head pose, illumination, facial expression, occlusion and aging. This thesis focuses mainly on the head pose problem and provides a reliable solution.

This introduction begins with an explanation of the objectives for this study in Section 1.1. Subsequently, an overview is presented in Section 1.2 which outlines the previous research works that are related to face recognition. We then describe the overview and structure of this thesis in Section 1.3.

1.1 Motivation

The difficulty of building a robust face recognition system (FRS) is that the face is a 3D object, which is illuminated from a variety of light sources and has many variations in appearance when it is projected onto a 2D image. Despite the extrinsic factors, the facial appearance of a single individual may be quite different due to deformations, expression, aging, facial hair and cosmetic, which is known as intra-personal variation. Large intra-personal variation may overtake inter-personal variation and thus lead to misclassification [3].

A robust and accurate face alignment is more critical among these factors according to the research in the past years [54, 46]. Face recognition technologies (FRT) such as Fisherfaces [3] and local appearance approach [24] show impressive performance if the faces have been manually aligned. However, in practical systems, automatic alignment, which is usually based on automatic eye localization, is far from precise. An imprecise localization of the eye centers with a possible deviation of only one or two pixels from their exact positions may cause false

classification. That means, the performance degradation mostly results from the incorrect alignment. Evidently, to solve the misalignment problem, a more accurate face alignment method should be developed. The robustness of the face representation and classification method to misalignment should be improved as well, which is beyond the range of this thesis.

Pose variation, especially out-of-plane rotation, greatly affects the alignment quality [4]. Self-occlusion occurs in this case, thus part of the face texture is occluded by the face itself while the frontal part may be stretched. Apparently, the appearance differs from the frontal view and recognition will not work at all. The problem can be solved by introducing profile or semi-profile views of the faces. However, the profile examples of the faces are not always available or sufficient to model the feature space for profile faces. Furthermore, face alignment in this case becomes tricky because the eye localization becomes even more inaccurate and in the worst case the eyes can not be detected at all. Another problem is that, alignment based on eye centers may not work as the eye distance can not directly indicate the size of the face.

Pose correction or pose normalization is studied to mitigate the pose variations in a way that a profile face is transformed to a frontal view. The transform in 3D space is linear but in 2D space it is non-linear. Although it is more intuitive to apply a 3D transform, alignment in 3D space such as 3D morphable model (3DMM) [7] requires much more computational effort which is not feasible for real-time systems. Hence in this thesis, the 2D active appearance model (AAM) fitting for face alignment is studied.

1.2 Previous work

This section gives an overview of previously developed face alignment approaches as well as face recognition systems. The first part concentrates on methods for face detection, which is an important step before face alignment. Some standard alignment approaches are described in the second part. Finally, different face recognition approaches are presented.

1.2.1 Face detection

As defined in [67], the goal of face detection is to "*determine whether or not there are any faces in the image and, if present, return the image location and extent of each face.*" Many factors pose challenge to face detection including: pose, facial expression, occlusion and image conditions, etc. A problem closely related to face detection is face localization, where the task is to return the locations and scales of faces on an image that contain faces.

In [67], face detection methods are classified into four categories: Knowledge-based, feature invariant, template matching and appearance-based approaches. Knowledge-based methods are developed based on rules derived from knowledge of human faces, such as the symmetry of a face and the relative distances and positions between facial features. This method showed poor performance in [65] despite its intuitiveness.

Feature invariant methods intended to find invariant features of faces for detection. These methods were based on the belief that there must exist properties or features on faces that are invariant to pose and illumination variations, since we, as human beings, can easily detect faces in different conditions. These invariant features can be low-level features like edges, skin color and motion, the position of eyes, nose and mouth and the geometric relations between them, for example.

Human skin color has been used and proven to be an effective feature for face detection and tracking. Different ethnical people may have different skin colors, but the major difference lies in their intensity rather than their chrominance [66]. Numerous approaches have been proposed to build a skin color model. Separating skin and non-skin colors by using a piecewise linear decision boundary was considered as the most simple method. For example, Chai and Ngan [9] proposed a face detection (segmentation) algorithm for a videophone application in which a fixed-range skin color map in the $CbCr$ plane is used. With carefully chosen thresholds, [Cr_1, Cr_2] and [Cb_1, Cb_2], a pixel is classified to have skin tone if its value (Cr, Cb) fall within the ranges, i.e. $Cr_1 \leq Cr \leq Cr_2$ and $Cb_1 \leq Cb \leq Cb_2$. More sophisticated classifiers were also investigated such as Bayesian classifier, where the class-conditional probability density functions (pdfs) of skin and non-skin colors can be estimated by using histograms [40] or mixtures of Gaussians [33]. Unfortunately, systems that rely on color alone are usually not robust against illumination changes and near skin color backgrounds.

The state-of-the-art approach for face detection was proposed by Viola and Jones [64]. The system is able to achieve high detection rates with relatively low computational effort, which is a great improvement compared to the previous systems. Partially motivated by the work of Papageorgiou et al. [51], they used a set of Haar like features to verify the existence of a face. In order to compute these features very rapidly at many scales they introduced the integral image representation for images. Instead of selecting discriminative features by some heuristics, they used a modified Adaboost [29] classifier to iteratively select important features from a huge set of potential features. Several of those classifiers are combined in a "cascade" which allows background regions to be pruned early while concentrating on more promising face-like regions. The false detections of one classifier are used to train its successor in the cascade. This framework can be easily extended for general object detection because the fea-

tures are selected automatically without using any structural knowledge about the objects.

The template matching methods usually use a standard face pattern which is manually predefined or parameterized by a function. The existence of a face is determined by the correlation between the template and the inspected image patch. A representative example approach is deformable templates which model facial features that fit an elastic model to facial features [68]. In this approach, facial features are described by parameterized templates. An energy function is defined to link edges, peaks, and valleys in the input image to corresponding parameters in the template. The best fit of the elastic model is found by minimizing an energy function of the parameters. However, the performance of this approach is sensitive to the initialization of the deformable template.

In contrast to the template matching methods where templates are predefined by experts, the "templates" in appearance-based methods are learned from example images. In this case, the relevant characteristics of face and non-face images are learned through statistical analysis and machine learning techniques and result in a form of distribution model or discriminant functions, which are consequently used for face detection.

Processing raw image intensities has a high computational cost. Usually, dimensionality reduction is carried out to improve detection efficiency. Turk and Pentland applied principal component analysis (PCA) to face detection as well as recognition [62]. They performed PCA on a training set of face images to generate so-called "eigenfaces" which span a subspace of the image space. Images of faces are projected onto the subspace and clustered. Non-face training images are projected onto the same subspace and clustered as negative training samples. The projections of non-face images appear more scattered and quite different from those of face images. To detect the presence of a face in a scene, the distance between an image patch and the face space is computed for all possible sub-patches in the image. The distance from face space is used as a measure of "faceness", and the calculated distances between all the patches and face space result in a "face map". A local minimum of the face map is considered as the detection of a face.

Rowley et al. [56] feed preprocessed face images into a set of retinally connected neural networks to learn the discriminant function that separates face and non-faces. To compensate for scale variations, the input image is downsampled into an image pyramid step by step, and a shifting-window slides over the different scales of the pyramid. Both the shifting window and the scaling require a high computational effort.

1.2.2 Face alignment

The output of face detection or face localization procedure can not be directly fed into the face recognition stage to extract feature vector, since the resulted face region is only coarsely localized and the discriminating features such as the salient facial features are not precisely aligned. The most simple but efficient alignment approach is to utilize the location of eye centers. An affine warp is performed according to the location, distance and rotation of left and right eyes. After warping, the eye centers are located in predefined coordinates. In [46], the importance of the warping step in face recognition is presented. However, utilizing the location of eye centers can not solve the problem of rotation in depth. Chai et al. [10] tackled this pose problem with the image synthesis strategy. They first estimate the face pose with a pose subspace algorithm, and divided the face region into three rectangles. An affine transformation is performed independently in the three rectangles and the transformation parameters are statistically learned from the correspondence information between the specific pose and the frontal face.

Active shape model (ASM) and active appearance model (AAM), proposed by Cootes et al. [15, 14], are two popular shape and appearance models for object localization. These two statistical model based approaches have attracted interest for many years and numerous extensions and improvements have been proposed.

In ASM [15], the local appearance model, which represents the local statistics around each landmark, allows for an efficient search to be conducted to find the best candidate point for each landmark. The search space is constrained by properly training a global shape model. The accuracy of local feature modeling may effect the precision of shape localization. AAM [14] combines constraints on both shape and texture in its characterization of facial appearance. The shape is optimized by minimizing the texture representation error. Once the model is fitted on an input image, the optimized model parameters are used for face recognition [21]. Recently, Guilllemaut et al. [35] investigated another AAM-based face alignment approach where the optimized model parameters were not used for direct face recognition, instead of that, they used a synthesized frontal appearance whose shape is normalized to a common mean shape. This approach has the advantage of preserving the textural information (moles, freckles, etc) contained in the original image; such information would be lost in the previous approach where the model parameters only represents the principal components of the appearance (low-pass filter equivalent). Moreover, the normalization of face shape solves the problem of rotation in depth by synthesizing a frontal view of the face. In this work, a similar pose normalization scheme is studied, however, the model shape is optimized with another fitting algorithm proposed by Matthew et al. [47]. Later, the model fitting algorithm as well as the pose

normalization algorithm will be presented in detail.

3DMM was successfully employed for face synthesis [6] as well as face registration. It is very similar to AAM in many respects. The major difference between them is that the shape of an AAM is defined in 2D whereas the shape component of a 3DMM is 3D. The 3D shape of a 3DMM is defined by a 3D triangulated mesh and in particular the vertex locations of the mesh. The appearance of a 3DMM is defined with a 2D triangulated mesh that has the same topology as the 3D mesh shape. Further differences between AAM and 3DMM are: (1) 3DMM is usually constructed to be denser; i.e. consist of more triangles. (2) Since it is defined in 3D, 3DMM can model self-occlusion (in terms of head rotation in depth), whereas 2D AAM cannot. In [7], face registration based on 3DMM fitting showed impressive recognition performance. However, since the 3D shape for a 3DMM is constructed much denser than in an AAM, fitting a 3DMM to an input image requires much more computational effort than fitting an AAM. Furthermore, model initialization in [7] was aided by user interaction.

1.2.3 Face recognition

The most popular holistic appearance based face recognition algorithm is called *eigenfaces* and was proposed by Turk and Pentland [62]. For a set of registered training face images, PCA is performed to reduce the dimensionality of the image space. The dimensionality reduction is achieved by finding a few orthogonal, uncorrelated principal components, which are the eigenvectors corresponding to the largest eigenvalues of the covariance matrix of the training set. In the context of face recognition, these are known as the eigenfaces. A new face can be expressed as a linear combination of these eigenfaces by projecting the image vector to the face space. Recognition is based on comparing the feature vectors in the face space. Fisherfaces [3], which is closely related to the eigenfaces approach, uses Fisher linear discriminate analysis (LDA) for dimensionality reduction. LDA seeks to find a linear transformation by maximizing the ratio of between-class variance to the within-class variance.

The holistic approaches, however, are very sensitive to occlusion and local variations such as changes in facial expression. To suppress the effects of the local variations on the whole appearance, local model based approaches were introduced. Modular eigenfaces [52] modeled salient local facial regions such as the eye and nose regions. Recognition is only based on the appearance of the local models and its performance is better than the holistic eigenfaces approach. However, the localization of the local regions is not an easy task, and requires precise facial feature detection. Erroneous detection of the facial features causes severe degradation in performance. A local appearance based approach has been proposed in [32], which divides the input face image into non-

overlapping blocks to perform eigenfaces locally on each block. Experiments showed that this method outperforms the standard holistic eigenfaces approach under variations of expression and illumination. Based on this idea, another local appearance-based approach was proposed in [24], which performs the discrete cosine transform (DCT) on local blocks to reduce dimensionality. The DCT is data independent, which means that the transformation bases are independent of the training data. Experiments showed that this approach outperforms the global version as well as other well known holistic approaches such as eigenfaces and Fisherfaces. Further details are given in Section 2.3 as this thesis is based on this approach.

1.3 Thesis overview

In this thesis, we investigated a face registration approach based on AAM fitting. We generated a single AAM to model shape variations of face rotation in pitch and yaw angles, so that pose information was obtained after model fitting. The pose of an input face was normalized with the pose information obtained from the fitted model parameters and a frontal view of the input face was synthesized. The modified histogram fitting approach was employed to mitigate illumination variations. To initialize the model more precisely and robustly against cluttered backgrounds a progressive model fitting scheme was used. A local appearance-based face recognition approach was performed on the fitted and pose-normalized face images.

We intensively conducted three experiments to evaluate the AAM-based face registration approach. The first experiment was designed to evaluate the pose correction based on AAM fitting in still images. The results showed a significant improvement in face recognition performance compared to the previous affine-based registration approach, which again demonstrated the importance of pose correction for face recognition. We also compared our local appearance-based face recognition approach with two well known holistic approaches. The local appearance-based approach significantly outperformed the holistic approaches and it was more robust against the error introduced by AAM fitting and face synthesis. The second experiment evaluated the eye localization with AAM fitting. Face tracking with AAM fitting was also evaluated on a video database for open set face recognition. A modified distance from feature space metric was employed to assess the quality of fitting on a single frame. Open set face recognition was performed on the successfully registered frames. The experimental results showed that both pose correction and registration quality assessment improved the recognition performance.

The remainder of the thesis is organized as follows. In Chapter 2, we describe the principle algorithms we used in this study. Then, we explain the pose and

illumination invariant face registration in Chapter 3. Experimental results are presented and discussed in Chapter 4. Finally, in Chapter 5 and 6, conclusions and future work are given.

2 Basic principles

This chapter describes the fundamental theories and related techniques that are employed in this work. The first sections will focus on AAM and the model fitting algorithm. Then the local appearance-based face recognition approach is explained.

2.1 Active appearance model

In this section, the basic information about active appearance models is provided. The statistical models of shape and texture will be first described, and consequently the model combination is discussed.

2.1.1 Background

Active appearance models (AAMs), are non-linear, generative, and deformable template-based models of a certain visual phenomenon [14]. They are closely related with the concepts of active blobs [57], and morphable models [7]. A precise definition of a deformable template model can be found in [28]:

Definition 1. *A deformable template model can be characterized as a model, which under an implicit or explicit optimization criterion, deforms a shape to match a known object in a given image.*

These deformable template models were developed based on the active contour model - known as Snakes - which was originally proposed by Kass et al. [41]. Snakes use a set of outline landmarks to delineate an object outline in a 2D image. The optimal displacement of the outline landmarks are achieved by minimizing an energy function that associates to a sum of internal and external energy of the current contour.

Later, Cootes et al. [15] proposed the active shape models (ASM) where shape variability is learned through observation. This is a more general approach and meanwhile specificity is also preserved through statistical modeling with principal component analysis. As a direct extension of the ASM approach, the active appearance models [14] were later proposed again by Cootes et al. Besides shape information, the textual information, i.e. the pixel intensities across the

Figure 2.1: The three steps of handling shape and texture in AAMs.

object, is included into the model. Further extensions and improvements were investigated in [16, 17].

The active blob [57] approach was proposed as a related parallel work to AAM. Instead of PCA, a finite element model (FEM) was employed to model shape variation. Moreover, it did not model the dynamic textual variation while the AAM approach optimized shape as well as textual parameters simultaneously.

Further information on deformable template models can be found in [5, 48].

2.1.2 Statistical model formulation: shape and appearance

As described above, AAM is modeled with shape geometry and texture information. To obtain a better understanding of the model, we first start by giving a definition of shape and texture [60]:

- Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object.
- Texture is the pixel intensities across the object in question (if necessary after a suitable normalization).

The construction of the model involves three major steps. As usual for data processes, the fist step is the data acquisition. When the data is ready, a suitable normalization follows. After that step the data is ready to be analyzed and described in terms of statistical models. The flow chart of the process is shown on Figure 2.1.

To stress the coherence between shape and texture handling, the steps are specified below.

Capture :

Shape Captured by defining a finite number of points on the contour of the object in question.

Texture Captured by sampling with a suitable image warping function.

Normalization :

Shape Brought into a common framework by aligning shapes with respect to position, scale and orientation using a Procrustes analysis.

Texture Removing global linear illumination effects by normalization.

Statistical Analysis :

Shape and Texture Principal component analysis is performed to achieve a constrained and compact description

Basically, there are just two types of linear shape and appearance models. The first type of models model shape and appearance independently, we refer to this type of models as *independent shape and appearance models*. The second set of models combine shape and appearance parameters into a single set of linear parameter which will be referred as *combined shape and appearance models*. In the next two subsections, the *independent AAMs* and the *combined AAMs* will be described separately.

2.1.2.1 Independent AAMs

As the name suggests, independent AAMs model shape and appearance (texture) separately. Shape and texture are necessarily normalized before statistical modeling with PCA.

Shape The first impression of the word "shape" might be a contour or an outline of an object. As defined before, the term shape is invariant to Euclidean transformations. One way to describe a shape is by locating a set of landmarks on the outline. In the context of AAM, the landmarks compose a mesh and their locations specify the vertex locations of the mesh. A v-point shape in k dimensional space can be represented by a kv dimensional vector. As we only consider 2D shapes in this thesis, the AAM shape s that makes up the mesh can be represented mathematically as a $2v$ dimensional vector:

$$s = [x_1, y_1, x_2, y_2, \ldots, x_v, y_v]^T. \tag{2.1}$$

Figure 2.2 shows an example face shape represented with 58 landmarks. The acquisition of these landmarks is usually done by manually placing several points along the contours of the salient facial features and the outline of the face. Annotating these landmarks on hundreds of images is a cumbersome work. Often, noise may be introduced which leads to imprecise modeling.

As defined before, location, scale and rotational effects need to be filtered out to obtain a true shape representation. This is carried out by translating,

Figure 2.2: A face shape represented with 58 landmarks.

rotating, and scaling all shapes to a common coordinate framework. The well-known Procrustes analysis [31] is used to obtain such a coordinate framework iteratively. The average of the N aligned shapes can be estimated as:

$$s_0 = \frac{1}{N} \sum_{i=1}^{N} \tilde{s}_i,$$

where $\tilde{s}_i$ denotes the i^{th} aligned shape. After alignment, the only difference of these set of shapes is the inter-point correlation. To handle the redundancy in multivariate data, PCA is used as a classical statistical method. PCA is also known as the *Karhunen-Loève transform*. It delivers the new axes ordered according to their variance. This can be easily understood via graphical illustration. In Figure 2.3, the two principal axes of a two dimensional data set are plotted and scaled according to the amount of variation that each axis explains. PCA is usually used as a dimensionality reduction method, by projecting a set of multivariate samples into a subspace constrained to explain a certain percent of the variation in the original data. For example, the 2D points in Figure 2.3 can be projected upon the first (largest) axis while the second principal axis is discarded.

Shape variation is described through performing PCA as an eigen-analysis of the covariance matrix of the aligned shapes. The covariance matrix can be given

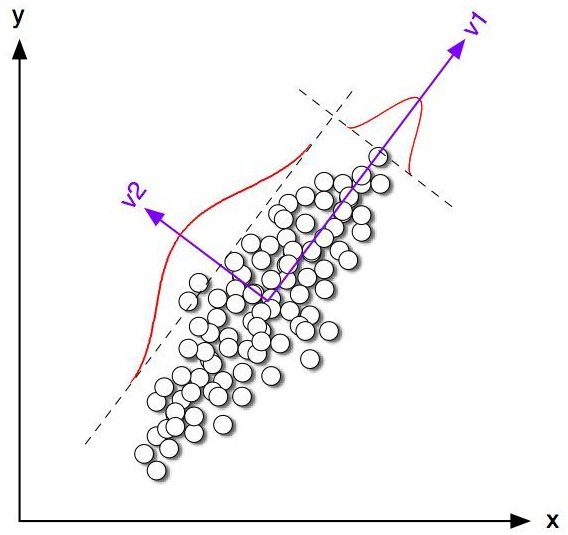


Figure 2.3: Data points are projected on the principal axes through PCA transform

as:

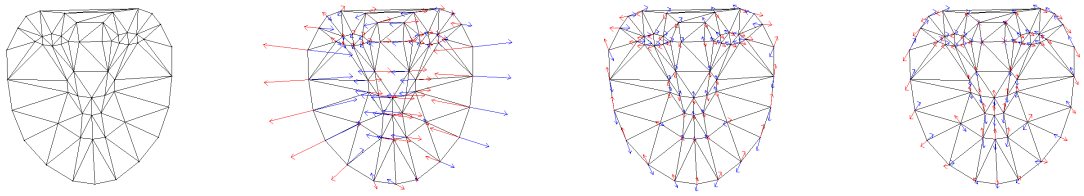
$$\Sigma_s = \frac{1}{N}\sum_{i=1}^{N}(\tilde{s}_i - s_0)(\tilde{s}_i - s_0)^T \tag{2.2}$$

The principal axes of the $2v$ dimensional shape vectors can be given as the n eigenvectors, s_i, of the covariance matrix, which correspond to the n largest eigenvalues of the covariance matrix. A shape instance s can then be expressed as a mean shape s_0 plus a linear combination of the n eigenvectors (eigenshapes):

$$s = s_0 + \sum_{i=1}^{n} p_i s_i \tag{2.3}$$

where p_i is parameter for the i^{th} shape component. The point representation of shape has now been transformed into a modal representation, where the modes are ordered according to their deformation energy, i.e. the percentage of variation that they explain.

The question is, up to how many modes should be retained. The small-scale variations in the model are considered as noises, which can be discarded. To compromise between the accuracy and the compactness of the model, we retained 95% of the shape variation which result in a 15 dimensional eigenspace. This is a rather substantial reduction since the original shape space had a dimensionality of $2 \times 58 = 116$. An example of face shape model is shown in Figure 2.4. On the left of the figure, the triangulated mean shape s_0 is plotted. In the remainder of the figure, the mean shape s_0 is overlaid with arrows



s_0 s_1 s_2 s_3

Figure 2.4: The linear shape model of an independent AAM. The model consists of a triangulated base mesh s_0 plus a linear combination of n shape vectors s_i. The base mesh is shown on the left, and to the right are the first three shape vectors s_1, s_2, and s_3 overlaid on the base mesh.

corresponding to each of the first three shape vectors s_1, s_2, and s_3.

Texture To complete the model one must also take *appearance* into account in addition to *shape* in contrast to ASM. Appearance of an object can be understood as texture in computer graphics. Texture is the pixel intensities across the object in question (if necessary after a suitable normalization). As the shape is represented by landmarks, the texture information can be collected from the pixels between the landmarks. This is done by an image warping function, for example, the piecewise affine warp based on the Delaunay triangulation of the mean shape. Hence, to obtain textual information from the training set, the shape is warped to a reference shape (here refers to the mean shape), and the pixel values inside the shape is sampled with bilinear interpolation. Hereafter, the triangulated reference shape is called *base mesh* as shown in Figure 2.4(s_0). After piecewise warping, the texture value inside the base mesh can be given as a texture vector A, usually normalization is necessary to remove the influence from global linear changes in pixel intensities.

The texture vector A is also named as *shape free image* since the image pixels are all located inside the base mesh. The dimensionality of a texture vector is far more higher than a shape vector. A 100×100 gray-level shape free image may result in a texture vector of several thousands of dimensions. As in the shape modeling, for dimensionality reduction PCA is used again to remove the data redundancy which is caused by sort of spatial correlations between pixels. The PCA transformation on the shape normalized texture results in m eigenvectors that correspond to m largest eigenvalues of the covariance matrix. The appearance of a sample independent AAM is shown in Figure 2.5. The three rows in this figure plot the effect of varying the first three eigen-appearances. An in-

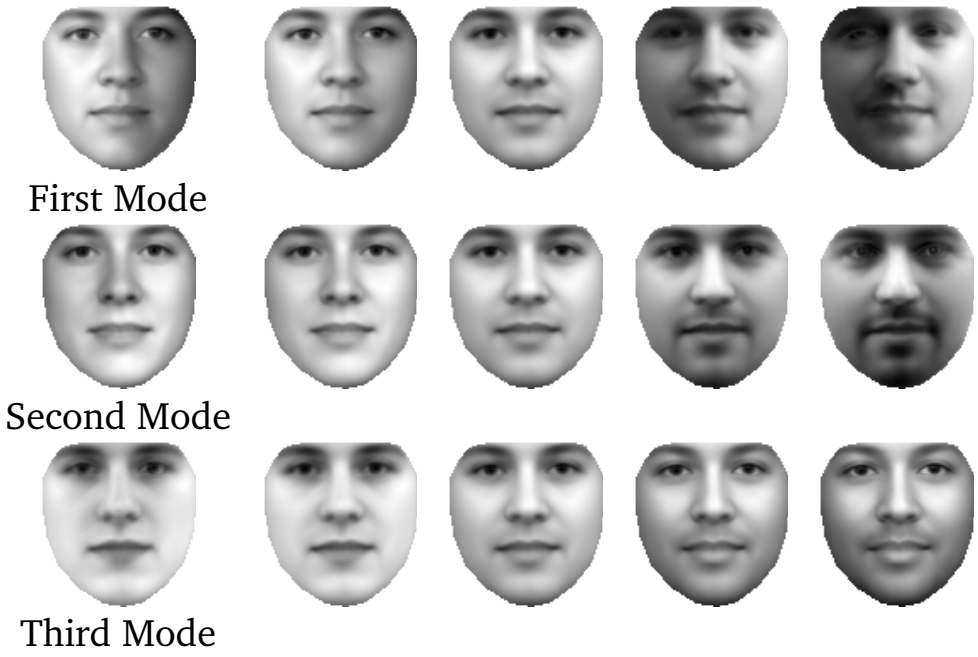


Figure 2.5: The linear appearance model of an independent AAM. Mean texture deformation using the first three principal modes

stance of model appearance is expressed as a linear combination of the mean appearance and the m eigen-appearances:

$$A = A_0 + \sum_{i=1}^{m} \lambda_i A_i,$$

where λ_i is the appearance parameter for the i^{th} component in the appearance model.

Model Instantiation We have modeled shapes and textures separately in two independent models. How to generate a single model instance from these two linear models? Given the AAM shape parameters $\mathbf{p} = (p_1, p_2, \ldots, p_n)^T$ we can use Eq. 2.3 to generate a shape instance of the AAMs. Similarly, we can generate an AAM appearance defined inside the base mesh s_0 by given appearance parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$. The AAM model instance with shape parameters $\mathbf{p}$ and appearance parameters λ is then generated by warping the appearance A from the base mesh s_0 to the model shape s using the warping function $\mathbf{W}(\mathbf{x}; \mathbf{p})$. This process is illustrated in Figure 2.6 for instantiated values of $\mathbf{p}$ and λ where the final AAM model instance is denoted as $M(\mathbf{W}(\mathbf{x}; \mathbf{p}))$.

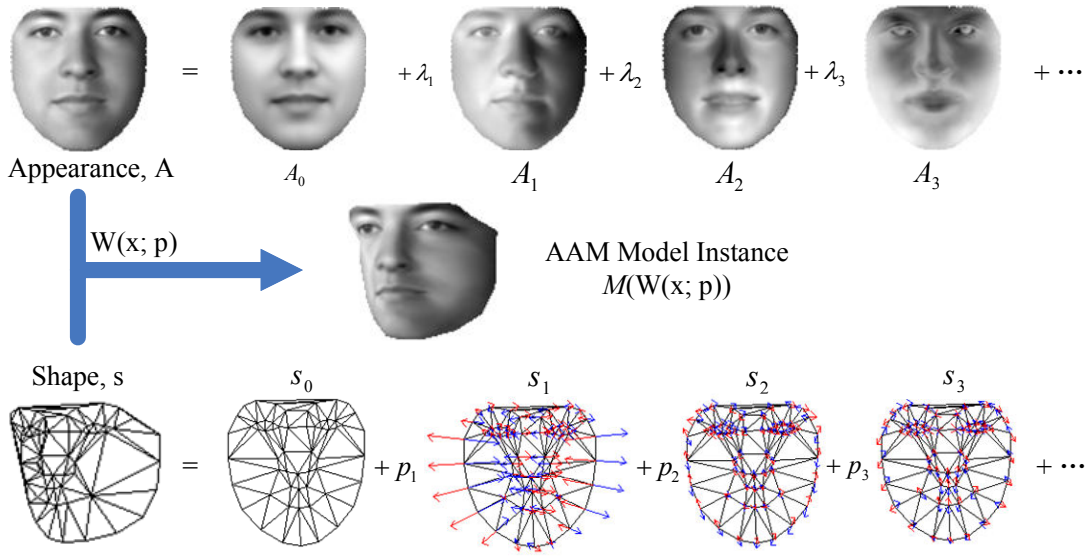


Figure 2.6: Model instance of an independent AAM. The shape parameter $\mathbf{p} = (p_1, p_2, \ldots, p_n)^T$ are used to compute the model shape s and the appearance parameters $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ are used to compute the model appearance A. The model appearance is defined in the base mesh s_0. The pair of meshes s_0 and s define a piecewise affine warp from s_0 to s which is denoted as $\mathbf{W}(\mathbf{x}; \mathbf{p})$. The final AAM model instance, denoted $M(\mathbf{W}(\mathbf{x}; \mathbf{p}))$, is computed by warping the appearance A from s_0 to s using $\mathbf{W}(\mathbf{x}; \mathbf{p})$.

2.1.2.2 Combined AAMs

While independent AAMs have separate shape $\mathbf{p}$ and appearance λ parameters, *combined* AAMs just use a single set of parameters $\mathbf{c} = (c_1, c_2, \ldots, c_l)^T$ to parameterized both shape and appearance in a compact way. A third PCA is performed on the shape and appearance parameters to remove the correlation between them. The concatenated shape and texture parameter $\mathbf{b}$ can be represented by the combined model parameter $\mathbf{c}$:

$$\mathbf{b} = \Phi_c \mathbf{c} \tag{2.4}$$

where Φ_c denotes a set of eigenvectors. Due to the linear nature of the model, the concatenated shape and appearance parameters are easily obtained through:

$$\mathbf{b} = \begin{pmatrix} W_s b_s \\ b_A \end{pmatrix} = \begin{pmatrix} W_s \Phi_s^T (s - s_0) \\ \Phi_A^T (A - A_0) \end{pmatrix} \tag{2.5}$$

where Φ_s and Φ_A are sets of eigenvectors for shape and appearance respectively, and W_s is a diagonal matrix providing a suitable weighting between pixel distances and pixel intensities. A complete model instance of combined AAMs including shape s and texture A, can be generated using the model parameter $\mathbf{c}$.

$$s = s_0 + \Phi_s W_s^{-1} \Phi_{\mathbf{c},s} \mathbf{c} \tag{2.6}$$

$$A = A_0 + \Phi_A \Phi_{c,A} \mathbf{c} \tag{2.7}$$

where

$$\Phi_c = \begin{pmatrix} \Phi_{c,s} \\ \Phi_{c,A} \end{pmatrix}. \tag{2.8}$$

The combined model is more compact than the independent model, which needs less parameters to represent the same visual phenomenon to the same degree of accuracy. Therefore, fitting may be more efficient with less parameters to be optimized. However, this coupling also has some disadvantages, for example, the assumption that the eigen-shapes s_i and eigen-appearances A_i are respectively orthonormal is no longer correct. Moreover, the choice of model fitting algorithms is restricted.

2.2 AAM fitting

In this thesis, we adopted a fitting algorithm based on independent AAMs. The basic algorithm is called "inverse compositional (IC) image alignment" proposed by Matthews and Baker [47]. Algorithms based on combined AAMs will not discussed here, the reader is referred to [14] in case of interest.

2.2.1 Fitting goal

Given an input image $I(\mathbf{x})$, the goal of fitting an AAM is to optimize the shape $\mathbf{p}$ and appearance λ parameters so that the model instance $M(\mathbf{W}(\mathbf{x};\mathbf{p})) = A(\mathbf{x})$ is similar in appearance to the input image. Thus the fitting criterion can be naturally defined as minimization of the error between the input image and the model instance. This error is computed in the coordinate frame of the AAM, i.e. the base mesh s_0. If $\mathbf{x}$ is a pixel in s_0, then the corresponding pixel in the input image I is $\mathbf{W}(\mathbf{x};\mathbf{p})$. At pixel x the AAM has the appearance $A(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})$. At pixel $\mathbf{W}(\mathbf{x};\mathbf{p})$, the input image has the intensity $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$. The fitting process is to minimize the sum of squares of the differences between these two quantities:

$$\sum_{\mathbf{x}\in s_0} \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})\right]^2 \tag{2.9}$$

where the sum is performed over all pixels $\mathbf{x}$ in the base mesh s_0. Minimizing the expression in Eq. 2.9 is considered as solving a non-linear least square problem. The shape parameters $\mathbf{p}$ as well as the appearance parameters λ are optimized simultaneously. The *error image* defined in the coordinate frame of the AAM can be denoted as follows:

$$E(\mathbf{x}) = A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})). \tag{2.10}$$

To compute such an error image, the input image I is first backwards warped onto the base mesh s_0 with warp $\mathbf{W}(\mathbf{x};\mathbf{p})$. The resulting shape free image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ is then subtracted from the model appearance $A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(i)$ and finally the result is stored in E.

2.2.2 Inverse compositional image alignment

The most natural way of solving a nonlinear least squares problem is to use a standard gradient descent optimization algorithm. These algorithms are analytically derived and the convergence properties are well understood. However, these gradient descent algorithms are very slow because for each iteration the partial derivatives, Hessian, and gradient direction need to be recomputed. The more efficient alternative fitting algorithms are mostly based on a simple assumption that there is a *constant* linear relationship between the error image $E(\mathbf{x})$ and the *additive* increments to the shape and appearance parameters:

$$\Delta p_i = \sum_{\mathbf{x}\in s_0} R_i(\mathbf{x}) E(\mathbf{x}) \qquad and \qquad \Delta\lambda_i = \sum_{\mathbf{x}\in s_0} S_i(\mathbf{x}) E(\mathbf{x}) \tag{2.11}$$

where $R_i(\mathbf{x})$ and $S_i(\mathbf{x})$ are constant images defined on the base mesh s_0. Constant means that $R_i(\mathbf{x})$ and $S_i(\mathbf{x})$ do not depend on p_i or λ_i. They are obtained during the time of modeling by adopting linear regression. Although this simple assumption leads to less computational cost for the additive increments to model parameters, its imprecision sacrificed the accuracy of fitting [47].

As a gradient-based image alignment algorithm, the inverse compositional (IC) algorithm is originated from the forwards-additive algorithm which is known as the Lucas-Kanade algorithm [45]. The goal of the Lucas-Kanade algorithm is to find the locally "best" alignment by minimizing the sum of squared differences between a constant template image $A_0(x)$ and an example image $I(x)$ with respect to the warp parameters $\mathbf{p}$:

$$\sum_{\mathbf{x}} \left[A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2. \tag{2.12}$$

Similar to Eq. 2.9, $\mathbf{W}(\mathbf{x};\mathbf{p})$ is a warp that maps the pixels $\mathbf{x}$ from the template image to the input image and has parameters $\mathbf{p}$. Thus the warped image $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$ has the same number of pixels as the template. For a given initial estimate of $\mathbf{p}$, the Lucas-Kanade algorithm solves for increments to the parameters $\Delta\mathbf{p}$ iteratively; i.e. minimize:

$$\sum_{\mathbf{x}} \left[A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p}))\right]^2 \tag{2.13}$$

with respect to $\Delta\mathbf{p}$ and the update $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Later in [58], Shum et. al extended the forwards-additive algorithm to the forwards-compositional algorithm. The *compositional* framework computes an *incremental warp* $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$ to be composed with the current warp $\mathbf{W}(\mathbf{x};\mathbf{p})$. The minimization is over:

$$\sum_{\mathbf{x}} \left[A_0(\mathbf{x}) - I(\mathbf{W}(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p})\right]^2, \tag{2.14}$$

and the update step involves $composing$ the incremental and current warp:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}). \tag{2.15}$$

As a further modification of these *forwards* algorithms, the *inverse computational* algorithm reverses the roles of the template and example image. Rather than computing the incremental warp with respect to $I(\mathbf{W}(\mathbf{x};\mathbf{p}))$, it is computed with respect to the template $A_0(\mathbf{x})$. The incremental warp is then estimated in the opposite direction since the roles of the images are reversed. With the roles reversed, the minimization problem can be formulated as:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{W}(\mathbf{x};\Delta\mathbf{p}))\right]^2, \tag{2.16}$$

and the current warp $\mathbf{W}(\mathbf{x};\mathbf{p})$ is updated by composing the reversed incremental warp:

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}. \tag{2.17}$$

Taking the Taylor series expansion of Eq. 2.17 gives:

$$\sum_{\mathbf{x}} \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{W}(\mathbf{x};\mathbf{0})) - \nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 \tag{2.18}$$

The warp with parameter $\mathbf{p} = \mathbf{0}$ is the identity warp; i.e. $\mathbf{W}(\mathbf{x};\mathbf{0}) = \mathbf{x}$. So the term $A_0(\mathbf{W}(\mathbf{x};\mathbf{0}))$ is the template $A_0(\mathbf{x})$. The gradient of the template A_0 and the *Jacobian* $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{0})$. The solution to this least squares problem is:

$$\Delta\mathbf{p} = H^{-1} \sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x};\mathbf{p})) - A_0(\mathbf{x}) \right] \tag{2.19}$$

where H is the Gauss-Newton approximation to the *Hessian* matrix:

$$H = \sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]. \tag{2.20}$$

The term $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the *steepest descent images*. Since the template A_0 is constant and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is always evaluated at $\mathbf{p} = \mathbf{0}$. The calculation of the steepest descent images and the Hessian matrix H is independent of the parameter $\mathbf{p}$, hence these steps can be pre-computed outside the updating iterations. The result is a very efficient image alignment algorithm. Figure 2.7 shows the required pre-computation steps and the steps inside the loop of the algorithm.

2.2.3 Fitting AAM with inverse compositional algorithm

Remember the goal of AAM fitting is to minimize the expression in Eq. (2.9) simultaneously with respect to the shape parameter $\mathbf{p}$ and appearance parameter λ. The linear appearance variations of the model should be considered in contrast to the constant template described in the last section. The "Project-Out" inverse compositional algorithm was proposed in [47], where the shape parameters $\mathbf{p}$ are found through non-linear optimization in a subspace in which the appearance variation can be ignored. The appearance variation is "projected out" from the *steepest-descent images* $SD_{ic} = \nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ by computing:

$$SD_{po}(\mathbf{x}) = SD_{ic} - \sum_{i=1}^{m} \left[\sum_{\mathbf{x} \in s_0} A_i(\mathbf{x}) SD_{ic}(\mathbf{x}) \right] A_i(\mathbf{x}). \tag{2.21}$$

The inverse compositional algorithm

Pre-compute:

(P1) Evaluate the gradient ∇A_0 of the template $A_0(\mathbf{x})$

(P2) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$

(P3) Compute the steepest descent images $\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}$

(P4) Compute the Hessian matrix using Eq. (2.20)

Iterate Until Converged:

(I1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$

(I2) Compute the error image $I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{x})$

(I3) Compute $\sum_{\mathbf{x}} \left[\nabla A_0 \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - A_0(\mathbf{x})\right]$

(I4) Compute $\Delta \mathbf{p}$ using Eq. (2.19)

(I5) Update the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$

Figure 2.7: The inverse compositional algorithm. The time consuming steps are performed in a pre-computing step. The steps inside the iterations can be implemented efficiently.

The computation of the incremental parameter updates Δp and the Hessian matrix is then carried out by replacing the steepest descent images SD_{ic} with the "Project-Out" SD images SD_{po} in Eq. (2.19) and (2.20).

The "Project-Out" IC algorithm is sufficient for fitting person specific AAMs [34]. Person specific AAMs contain less variation in appearance since the model is generated for a single specific person. However, in this thesis, the application scenario is more generic where the fitting should achieve reasonable accuracy on all possible faces including the "unseen faces". The *simultaneous inverse compositional (SIC)* algorithm was proposed in [34] to improve the performance of fitting generic AAMs.

The SIC algorithm operates by iteratively minimizing,

$$\sum_{\mathbf{x}} \left[A_0(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})) + \sum_{i=1}^{m} (\lambda_i + \Delta \lambda_i) A_i(\mathbf{W}(\mathbf{x}; \Delta \mathbf{p})) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))\right]^2 \qquad (2.22)$$

simultaneously with respect to $\Delta \mathbf{p}$ and $\Delta \lambda = (\Delta \lambda_i, \ldots, \Delta \lambda_m)^T$, and then updating the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$ and the appearance parameters $\lambda \leftarrow \lambda + \Delta \lambda$.

Combining the warp parameters $\mathbf{p}$ and the appearance parameters λ results in an $n+m$ dimensional column vector $\mathbf{q} = [\mathbf{p}, \lambda]^T$, and similarly the update $\Delta \mathbf{q}$ also becomes $[\Delta \mathbf{p}, \Delta \lambda]^T$. The steepest-descent images are expanded to $n + m$

dimensions and denoted as follows:

$$SD_{sim}(\mathbf{x}) = (\nabla A \frac{\partial \mathbf{W}}{\partial p_1}, \ldots, \nabla A \frac{\partial \mathbf{W}}{\partial p_n}, A_1(\mathbf{x}), \ldots, A_m(\mathbf{x})) \tag{2.23}$$

where ∇A is defined as

$$\nabla A = \nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i. \tag{2.24}$$

Then the update to parameter $\mathbf{q}$ can be computed as

$$\Delta \mathbf{q} = -H_{sim}^{-1} \sum_{\mathbf{x}} SD_{sim}^{T}(\mathbf{x}) E_{app}(\mathbf{x}) \tag{2.25}$$

where H_{sim}^{-1} is the inverse of the Hessian matrix:

$$H_{sim} = \sum_{\mathbf{x}} SD_{sim}^{T}(\mathbf{x}) SD_{sim}(\mathbf{x}) \tag{2.26}$$

and E_{app} is the *error image* between the warped input image and the model instance:

$$E_{app}(\mathbf{x}) = I(\mathbf{W}(\mathbf{x}; \mathbf{p})) - \left[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})\right] \tag{2.27}$$

As shown in Eq. (2.23), the steepest descent images SD_{sim} are dependent on the appearance parameters λ so they are no longer constant as the "Project-Out" steepest descent images SD_{po} and have to be re-computed in every iteration. This makes the simultaneous algorithm less efficient compared to the "Project-Out" algorithm. The steps of the SIC algorithm are given in Figure 2.8.

Note that up to now, the *global shape normalizing transform* of shape is not taken into consideration for the simplicity of explanation. As we described before, the training shapes are normalized using an iterative Procrustes analysis that removes, for example, translation, rotation and scale differences across all training shapes. Because of this normalization, the shape vectors of the AAM do not model the 2D similarity transformation (translation, rotation and scale). However, the image data that the AAM is fitted to will be translated, rotated, and scaled by image formation process (i.e. during capture). To fit a "normalized" AAM to such data it is therefore necessary to incorporate a matching global shape transform to the AAM. This is achieved by parameterizing the global shape normalizing transform N with extra four parameters and composing the transform with the piecewise affine wrap $\mathbf{W}(\mathbf{x}; \mathbf{p})$. For details please refer [47].

The simultaneous inverse compositional algorithm

Pre-compute:

(P1) Evaluate the gradient ∇A_0 and ∇A_i for $i = 1, \ldots, m$

(P2) Evaluate the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $(\mathbf{x}; \mathbf{0})$

Iterate Until Converged:

(I1) Warp I with $\mathbf{W}(\mathbf{x}; \mathbf{p})$ to compute $I(\mathbf{W}(\mathbf{x}; \mathbf{p}))$

(I2) Compute the error image $E_{app}(\mathbf{x})$ (Eq. (2.27))

(I3) Compute the steepest descent images $SD_{sim}(\mathbf{x})$ (Eqs. (2.23))

(I4) Compute the Hessian H_{sim} using Eq. (2.26) and invert it

(I5) Compute $\sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) E_{app}(\mathbf{x})$

(I6) Compute $\Delta \mathbf{q} = -H^{-1}_{sim} \sum_{\mathbf{x}} SD^T_{sim}(\mathbf{x}) E_{app}(\mathbf{x})$

(I7) Update the warp $\mathbf{W}(\mathbf{x}; \mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x}; \mathbf{p}) \circ \mathbf{W}(\mathbf{x}; \Delta \mathbf{p})^{-1}$ and $\lambda \leftarrow \lambda + \Delta\lambda$

Figure 2.8: The simultaneous inverse compositional algorithm.

2.3 DCT-based local appearance face recognition

Appearance-based subspace approaches such as Eigenfaces have dominated the face recognition research during the last years. Experiments have shown that component based [37] and modular approaches [52], which use local regions of salient features, are superior to the holistic template-based approaches. But, the detection of salient features -i.e. eyes- is not an easy task. Moreover, erroneous detection of these facial features may severely degrades the performance. A local appearance based approach has been proposed in [32], which divides the input face image into non-overlapping blocks to perform Eigenfaces locally on each block. The experiments conducted in [32] showed that the proposed method outperforms the standard holistic Eigenfaces approach under variations of expression and illumination.

In this work, we apply the novel local appearance based approach using DCT. The underlying idea is to utilize local information while preserving the spatial relationships. In [23], the DCT is proposed to be used to represent the local regions. The DCT has been shown to be a better representation method for modeling the local facial appearance compared to PCA and the discrete wavelet transform (DWT) in terms of face recognition performance [23]. The discrete cosine transform for 2D input $f(x, y)$ is defined as:

$$C(u,v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right],$$

for $u, v = 0, 1, \ldots, N-1$, where

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u = 1, 2, \ldots, N-1 \end{cases}$$

And the 2-D inverse discrete cosine transform is defined as

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(u,v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right].$$

The corresponding DCT bases are shown in Figure 2.9. As can be seen from the top-left part of the basis functions, the $(0,0)$ component represents the average intensity value of the image. From the figure, it can be also noticed that the $(0,1)$ and $(1,0)$ components represent the average vertical and horizontal changes, and the $(1,1)$ component represents the average diagonal changes in the image. Lower order coefficients represent lower frequencies, whereas higher order coefficients correspond to higher frequencies.

Feature extraction from registered images using local appearance-based representation can be summarized as follows: The input image is scale into 64×64 pixels size and divided into blocks of 8×8 pixels size. Each block is then represented by its DCT coefficients. These DCT coefficients are ordered using a zig-zag scanning pattern [30] (see Figure 2.10). From the ordered coefficients, M of them are selected according to a feature selection strategy resulting in an M-dimensional local feature vector. Finally, the DCT coefficients extracted from each block are concatenated to construct the overall feature vector of the corresponding image.

2.4 K-NN classification

The k-nearest neighbor (NN) algorithm is an easy, efficient and important non-parametric classification algorithm. Theoretically if we have infinite sample points, then the density estimate converges to the actual density function. The classifier becomes Bayesian classifier if large number of samples are provided. But in the case of face recognition, we generally have limited number of instances for each face class, which is not enough for density estimation. The method of k-NN is often used in this case.

Usually, the Euclidean distance metric is used to determine the nearest k neighbors of the test sample. A special case of the k-NN is the 1-NN method, which just searches the nearest neighbor for a given feature vector f:

$$j = \arg\min_i \|f - f_i\|,$$

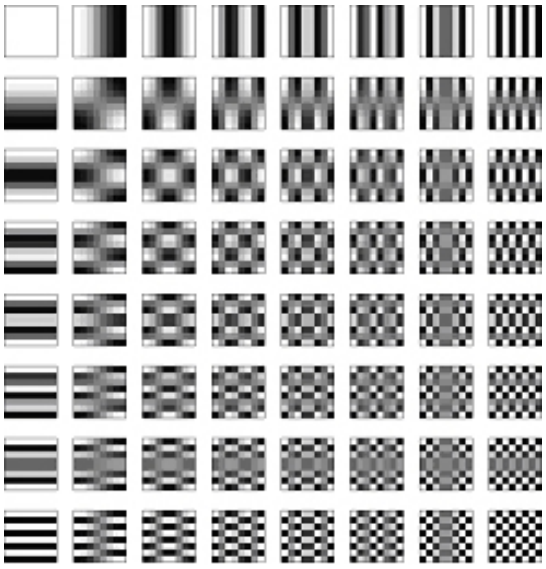


Figure 2.9: DCT basis functions for 8×8 pixel images, the origin is at the top left corner.

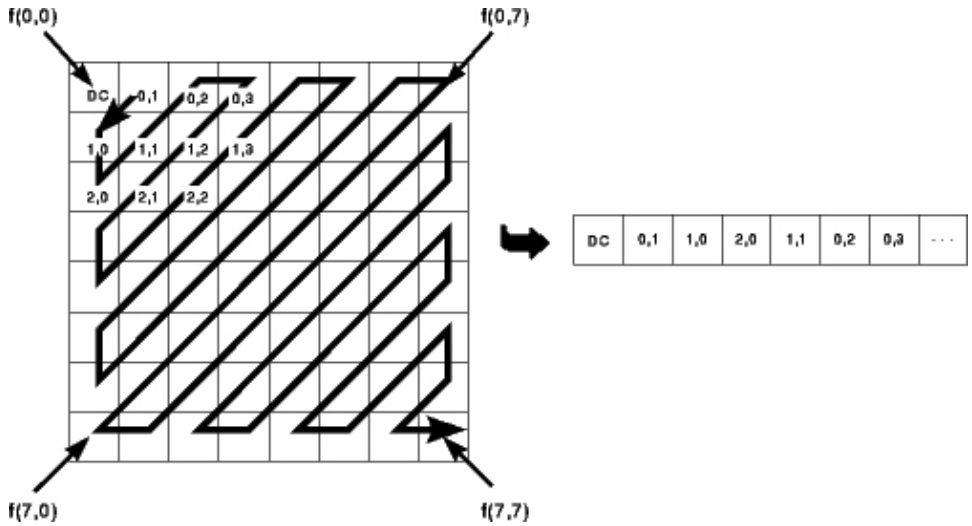


Figure 2.10: Zig-zag scanning, first coefficient represents average intensity value of the block.

then the nearest neighbor of f is the j^{th} feature vector in the training set. The euclidean distance measure corresponds to L2 norm. Other norms are also considered as distance norm, such as L1 norm which is defined as follows:

$$d_{L1} = \sum_{m=1}^{M} |f_{training,m} - f_{test,m}|,$$

where $f_{training,m}$ is the $m^{th}(m = 1, \ldots, M)$ coefficient in the training feature vector $f_{training}$. Similarly, $f_{test,m}$ is the m^{th} coefficient in the test feature vector f_{test}.

The distance metrics based on correlation and covariance are also discussed in [24] for nearest neighbor classifier. A detailed performance comparison between these distance metrics for our local appearance-based face recognition algorithm is also given in [22].

3 Methodology

After introducing the main theoretical foundations of this work, this chapter will present the design decisions for the face alignment task. First we start with building a generic active appearance model. The model initialization and robust model fitting against illumination changes will be described subsequently. In Section 3.4, pose normalization methods will be discussed. Finally, face recognition based on the proposed face alignment method is described.

3.1 Model building

As suggested in [34], the performance of person specific AAMs is substantially better than the performance of generic AAMs. This is because the person specific AAMs only model the variations in appearance of a single person across pose, illumination, and expression; while the generic AAMs also model identity of subjects that cause variations both in appearance and shape. The study in [34] shows that building person specific AAMs is far more easier than building generic AAMs. Moreover, person specific AAMs are more robust than generic AAMs when performing model fitting. The variation in shape across different identities effect the robustness of fitting generic AAMs. In order to reduce the effects of reconstruction error, more training images are needed to cover all possible variations in pose, illumination, expression and identity.

In this thesis, we collected a set of 649 images from four publicly available databases. The first one is the IMM face database [2], which contains 240 face images with frontal and semi-profile views from 40 subjects. This database is annotated with 58 landmarks for AAMs. The second is the ND1 database [12], which contains 953 face images with mostly frontal views from 273 subjects. From the ND1 database we selected 100 images from different subjects to model the variation of identity. The last two databases are the CMU PIE database [59] and the FERET database [55]. From the CUM PIE database we used 76 images from 76 different subjects to model the illumination component in appearance. The remaining images are taken from the FERET database, which contains a large number of different subjects with pose and expression variations. The example images selected from the four databases are displayed in Figure 3.1. Since the landmarks are manually placed on the training images, the labeling work is a time-consuming operation and is error-prone due to the accuracy lim-

itations of a manual operation, moreover, the interpretation of *correct* landmark locations are different. For example, if the same person annotates a given image multiple times, noticeable differences in labeling can occur. Also, interpretation for the definition of certain landmarks differs from person to person, especially the landmarks along the boundary of the cheek, where there is no distinct facial feature to rely on. The inconsistent labeling introduces noises in shape data and effects face modeling. The shape basis will not only model the inherent shape variation, but also the error of the labeling, which may significantly effect the performance of fitting.

To solve this problem, the data refitting method is employed as suggested in [34] and [43]. The idea is simple but effective. After an AAM is trained with a set of training images and manual labels, we perform fitting on the same training images using the SIC algorithm, where the manual labels are used as the initial location for fitting. This fitting generates new landmark positions for the training images. With these new landmarks, a new AAM is trained and fitting is applied again on the training images. This process iterates until there are no significant changes in landmark locations. Figure 3.2 plots the diagram for this data fitting process.

3.2 Initializing AAM fitting

After building a generic AMM, an instance of the model is defined by manipulating the global shape transform and model parameters. Fitting the model to unseen data is equivalent to finding a configuration of parameters that optimally fit the model to the unseen data. However, the gradient descent based AAM fitting scheme is inherently dependent on good initialization. Poor initialization of model pose and parameters makes the fitting easily stuck into local minimum and diverges away from the target. The most simple way to initialize the model is the brute-force method which iteratively tries out every possible configuration. However this method is very time-consuming since the number of possible initializations is huge. In [60] it has been suggested to perform AAM search in parallel with multiple different initialization parameters, i.e. the perturbed pose and model parameters. This method is less time-consuming than the brute-force searching though, it's still far from efficient especially for real-time applications.

Since the fitting target in this thesis is the human faces, the model can be initialized efficiently thanks to the fast and robust face detector based on Haar-like features. The Haar-cascade face detector [64] is the state-of-the-art face detector in recent years. The detector is able to achieve a reasonable high detect rate in a high frame processing rate. The face detector remove uninteresting clusters while detecting faces as foreground. The face rectangles as the output of the face detector can be considered as the initial position and size of a fitting

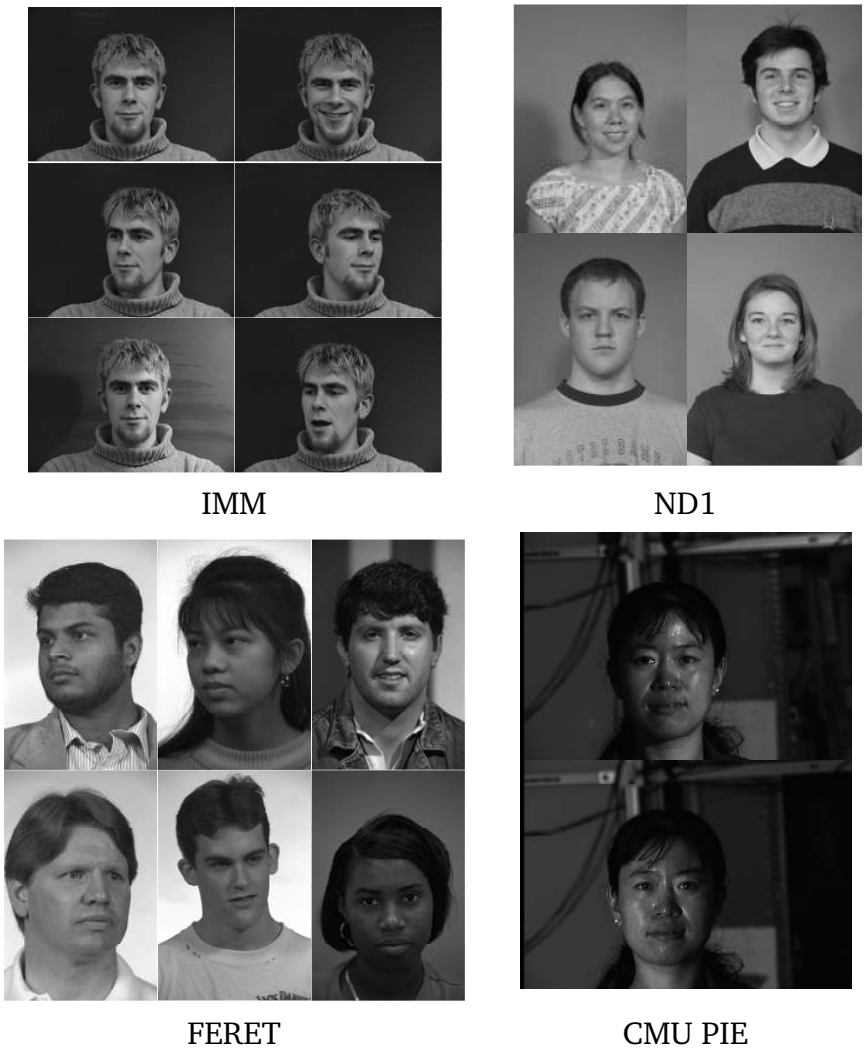


Figure 3.1: Sample images from IMM, ND1, FERET and CMU PIE face databases for training a generic model.

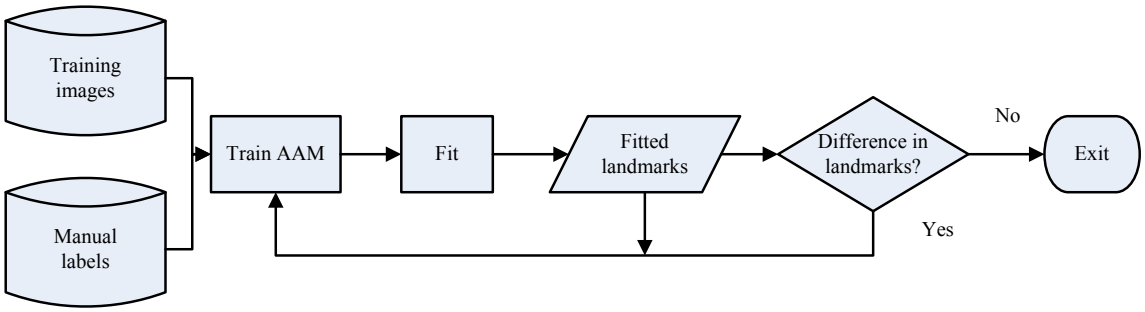


Figure 3.2: The data refitting schema. Iterative face model building and fitting using given set of training images. The process is terminated when there are no significant differences in landmarks.

model.

The detected face rectangles suffice the AAMs initialization most of the time when the face in question is near frontal and near upright. However, the detected face rectangles usually do not consistent in size even with minor change in face appearance. This makes the estimation of model size difficult and the fitting performance is unstable. To tackle this problem, more facial details are utilized to provide the fitting-process a better initialization.

For a given input image, we first try to detect the face regions using the Haar-cascade face detector. In the case that a face rectangle is detected, we perform further facial feature detection using the Haar-cascade detector again, where the Haar-like features are extracted on the specified facial features such as eyes and mouth. Since the facial features are not always detectable, we discuss 4 possible combinations:

- Only face but no facial features are detected. The model is initialized by placing the base mesh inside the face rectangle with proper scale and translate.
- Face and only eyes are detected. The model position is initialized according to the coordinates of the eye centers and the initial shape size is estimated according to the size of the face rectangle.
- Face and only mouth are detected. The model position is initialized according to the coordinates of the mouth center and the initial shape size is estimated according to the size of the face rectangle.
- Face and both eyes and mouth are detected. Estimate affine transform according to the facial features on the input image and the base mesh. And perform this affine transform on the base mesh so that the differences

in locations between the feature points on the base mesh and the input image are minimized.

This procedure initialize the pose of the model, the shape and texture parameters are set to be *zero* which means that the model is initialized with mean appearance A_0 and base mesh s_0, where the base mesh is scaled, rotated and translated to the estimated scale and position.

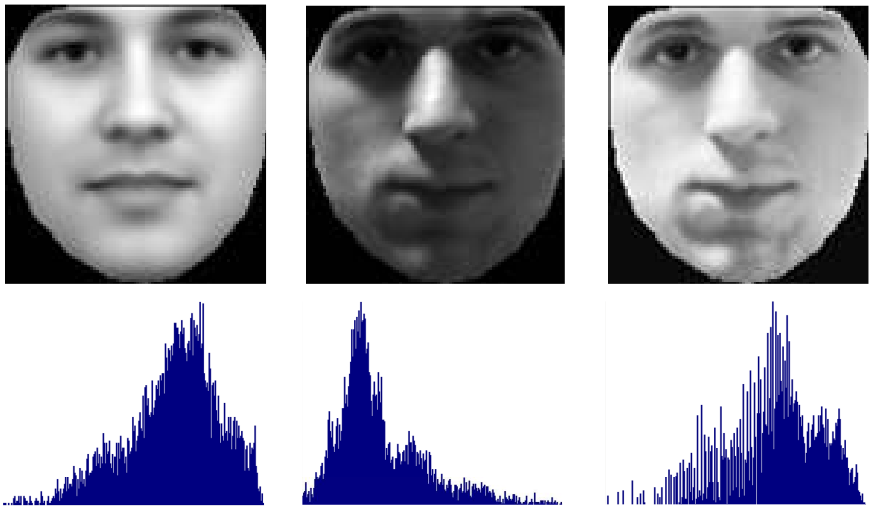
3.3 Robust fitting issues

Starting with the initialized model pose and parameters, the SIC searching algorithm is performed iteratively until the the algorithm converges or the iteration number reaches a predefined limit. The appearance of a face differs under different illumination conditions, which may affect the accuracy and robustness of model fitting. The cluttered background is also an important factor for robust fitting in case of a poor model initialization. The following subsections discuss these problems and present the possible solutions to solve such problems.

3.3.1 Fitting across illumination

Fitting AAMs on the images captured under different illumination conditions is an open problem. Although the AAMs model the illumination changes in appearance, fitting may still fail on the images with extreme illumination due to the large reconstruction error. Many approaches have been investigated to reduce the illumination effects in image alignment. In [18], the identity variability and the illumination variability were modeled separately. While fitting the model, they projected out the appearance variation introduced by illumination changes. The idea is similar to the "project-out" IC algorithm, where appearance variations are projected out while optimizing the shape parameters. Another common approach is histogram based illumination normalization. The global histogram equalization methods used in image processing for normalization only transfers the holistic image from one gray scale distribution to another. However, this global processing can not normalize the gray level distribution variations on a specified local face region. As suggested in [39], the histogram fitting algorithm (or histogram specification) is able to normalize a poorly illuminated image to a similar, well illuminated image. In this study, we adopted this histogram fitting algorithm for face illumination normalization.

The detail of the histogram fitting algorithm is described in [30]. The underlying idea of this algorithm is to transform the input image so that it has a particular histogram as specified. The particular histogram is extracted from a



(a) (b) (c)

Figure 3.3: Light normalization using histogram fitting: (a) Mean face and its histogram, (b) Test face and its histogram, (c) Normalized face and its histogram.

well illuminated image, here the mean face inside the AAM base mesh is used. The histogram of the mean face H_G is depicted in Figure 3.3.

We consider the most common case of the illumination problem where one side of the face is dark and the other side is bright. The problem is caused by unequal illumination on the left and right sides of the face. Hence the histogram fitting is performed independently to both side of the face. The face image is first split into two parts (left/right) and then the histogram of each window is extracted. The histogram of the left window is denoted as $H_l(i)$ and the histogram of the right window is denoted as $H_r(i)$. Both histograms will be fitted to the histogram H_G on the mean face. The two mapping functions corresponding to the left and right windows are denoted as: $f_{H_l \to H_G}$ and $f_{H_r \to H_G}$. The sudden discontinuity in illumination may introduce artifact as we switch from the left side of the face to the right side. To overcome this problem, a transition function is used to "average" the effects of $f_{H_l \to G}$ and $f_{H_r \to G}$ with a linear weighting. The weighted combination of the two mapping function yields a third mapping function $f_{H_{total} \to H_G}$ shown in Eq. 3.1.

$$f_{H_{total} \to H_G}(i) = leftness \times f_{H_l \to h_G}(i) + (1 - leftness) \times f_{H_r \to H_G}(i), \qquad (3.1)$$

The parameter leftness is varied as we move across the image from left to right, which is considered as a smooth function.

Figure 3.3 plots the employed illumination normalization method together with the corresponding histogram. The histogram fitting is performed on the

input image (first row, middle). The normalized image is plotted on the last column of the first row in Figure 3.3, where the histogram of the restored image is very close to the histogram of the reference image as expected.

3.3.2 Progressive fitting

As it is known, face alignment using AAM is sensitive to model initialization [19]. If the initial points do not start sufficiently close to the global minimum, in some cases the fitting algorithm may converge to a local minimum so that the face alignment using AAM does not provide precise results. As described before, the initial model shape and its pose is estimated by the detected face rectangle and facial features. However, the initial values for feature points in the contour of chin may be not close to global minimum, especially when the face in the input images differs from the base mesh in terms of pose or outline shape. In the case of complex background, large error in estimation of the feature points around the face outline will affect the localization of the other facial features points. Thus, localizing relatively stable inner facial feature points can be affected by the feature points in the outer chin area when one tries to localize the whole facial feature points at the same time.

To mitigate the influence of the poor initialized outline feature points in the chin area, we built another AAM model in which the landmark points in the chin area are removed. As plotted in Figure 3.4, the new model only contains the inner part of the face where the more stable facial features such as the eyes, eye brows, nose and mouth are modeled. The new simplified model is called the *inner face AAM* hereafter, while the original one is called the *whole face AAM*.

The fitting of the two models is splitted into two sequential stages. In the first stage, the inner face AAM is fitted to the inner part of a new face image. The initialization of the inner face AAM is similar to the method described in subsection 3.2. In the second stage, the initial shape parameter $\mathbf{p}$ of the whole face model is estimated from the obtained shape parameter $\hat{\mathbf{p}}$ in the first stage:

$$\mathbf{p} = R\hat{\mathbf{p}}, \tag{3.2}$$

here we assume R is constant and it is obtained by performing *linear regression* on the training data of the both models. A second AAM fitting is carried out to fit the whole face AAM to the face image using the initial parameters $\mathbf{p}$. The bi-stage AAM fitting provides robust fitting results in [42], because after the first stage fitting, the initialization of the whole face AAM is close to the global minimum which makes the final fitting efficient and effective. Figure 3.5 depicts the bi-stage AAM fitting scheme.

Figure 3.4: Inner face landmarks.

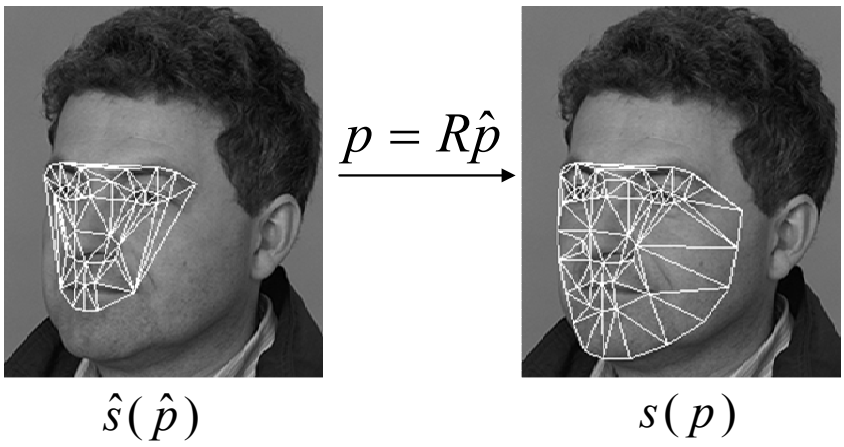


Figure 3.5: Progressive AAM fitting.

3.3.3 Fitting while tracking

In addition to aligning faces in still images, the face alignment in video sequences with AAM tracking is also studied in this thesis. In the context of AAM tracking in video sequences, the AAM fitting is performed on each frame by using the fitting results, i.e., the shape and appearance parameters, of the previous frame as the initialization of the current frame. As we addressed before, fitting a generic AAM to faces of an unseen subject can be hard due to the mismatch between the appearance of the facial images used for training the AAM and that of the testing video frames, especially when the lighting conditions differ. As demonstrated in [34], even using the proposed simultaneous IC algorithm, the fitting performance of a generic AAM is not as good as fitting a person specific AAM due to the reconstruction error of fitting on the unseen faces.

To improve AAM fitting in video sequences, Liu [44] extended the Simultaneous IC algorithm to enforce the frame-to-frame registration across video sequences. Similar to the SIC, the proposed algorithm minimizes the distance of the warped image and the generic AAM model during the fitting, but in the meanwhile, it also minimizes the distance between the current warped image and a model obtained from the warped images of previous video frames. They call their extended approach as "SIC fOr Video (SICOV)" algorithm.

Given a generic AAM and a video frame I_t at time t, the fitting goal of the SICOV algorithm is to minimize the following cost function:

$$\sum_{\mathbf{x}} \left[A_0(W(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{W}(\mathbf{x})) - I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2 + k \sum_{\mathbf{x}} \left[M_t(\mathbf{x}) - I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))\right]^2, \tag{3.3}$$

which is a weighted composition of two cost functions by a constant k. The first term is the fitting goal of the SIC algorithm (see Eq. 2.22). The second term is the sum of squared error between the current warped image $I_t(\mathbf{W}(\mathbf{x};\mathbf{p}))$ and the appearance information of the current subject from previous frames, $M_t(\mathbf{x})$. For simplicity, we just define $M_t(\mathbf{x})$ as the warped image of the video frame at time $t-1$:

$$M_t(\mathbf{x}) = I_{t-1}(\mathbf{W}(\mathbf{x};\mathbf{p}_{t-1})). \tag{3.4}$$

Further definition of $M_t(\mathbf{x})$ is also possible, for example, the weighted average of L previous warped images can be defined as the $M_t(\mathbf{x})$, where the weights are calculated by a decay factor [44]. The benefit of this term is clear, it presents the specific appearance information of the subject being fitted, which may not be modeled by the generic face models. This information can compensate the mismatch between the face models and the input images being fitted. However, if we only use the subject-specific model, the alignment error would propagate over time. The generic model is well suited for preventing the error propagation

and correcting the drifting. Thus during the fitting of each frame, both terms are served as constrains to guide the fitting process.

The detailed derivation of the SICOV algorithm is described in [44]. The different computation steps in comparison to the SIC algorithm are the calculation of the steepest-descent images and the error images. The steepest-descent images are formulated as following:

$$SD_{SICOV}(\mathbf{x}) = \big[(\nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i + k\nabla M_t)\frac{\partial \mathbf{W}}{\partial p_1}, \ldots, (\nabla A_0 + \sum_{i=1}^{m} \lambda_i \nabla A_i + k\nabla M_t)\frac{\partial \mathbf{W}}{\partial p_n}, A_1(\mathbf{x}), \ldots, A_m(\mathbf{x})\big], \quad (3.5)$$

where ∇M_t is the gradient of the term M_t, in this study, it is the gradient of the warped image at $t-1$. And the error image is denoted as:

$$E_{app}(\mathbf{x}) = I_t(\mathbf{W}(\mathbf{x};\mathbf{p})) - \big[A_0(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})\big] + k(I_t(\mathbf{W}(\mathbf{x};\mathbf{p})) - M_t(\mathbf{x})). \quad (3.6)$$

Note that in Eq. (3.5) and (3.6), extra computation is required because the previous warped appearance $M_t(\mathbf{x})$ is involved. However, as reported in [43], the average number of iterations in fitting a video sequence using SICOV algorithm is lower than using the original SIC algorithm. With the constraint of the subject-specific model, the fitting is more efficient and effective.

3.3.4 Re-initialization while tracking

After the fitting process with the AAM searching algorithm, we obtain a face texture A which is defined in the base mesh. A simple way to verify the result of the fitting is to check the residual error, which is also been considered as the stop criterion of the fitting iteration. The residual error indicates the reconstruction error of the eigenspace decomposition and the measure is referred as the "distance from feature space" (DFFS) in the context of "eigenfaces". However, the error of the AAM face fitting is composed of reconstruction error and search error. The residual error alone is not able to verify the quality of the fitting results. Eigenfaces, especially higher-order ones, can be linearly combined to form images which do not resemble faces at all. In this sense, the coefficients of the eigenfaces should also be taken into consideration. For this purpose another distance measure, the "distance in feature space" (DIFS), is introduced. We use the combined distance measure DFFS+DIFS as the indicator for the quality of fitting.

As defined in [49], the DFFS indicates the distance of the input feature vector from the face space, as illustrated in Figure 3.6. The formulation of the DFFS

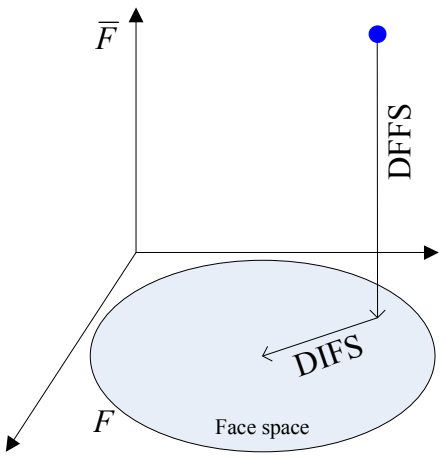


Figure 3.6: Distance from feature space(DFFS) and distance in feature space (DIFS).

for a given texture vector A is given as following:

$$DFFS(A) = \|A - (A_0 + \sum_{i=1}^{M} \lambda_i A_i)\|, \tag{3.7}$$

where A_0 is the mean appearance as defined before and λ_i is the coefficient corresponding to the i^{th}eigenface A_i. The DIFS measure indicates the distance from the projected input feature vector to the center of the face space. It is defined as the Mahalanobis distance between the vector λ and the center of the face space:

$$DIFS(A) = \lambda^T \Lambda^{-1} \lambda, \tag{3.8}$$

where Λ is the diagonal matrix of the eigenvalues corresponding to the M eigenfaces.

As expected, the distribution of the eigenface coefficients λ_i and the residual value ε is a multi-variate Gaussian with different variance σ^2. The variance of the eigenface coefficients is the corresponding eigenvalue. However, the extreme outliers will be rejected as "non-face" if the variance of the data in each eigenspace is used as the sigmas in the Gaussian distribution. These outliers lies quite far away from the face cluster, however, they are still valid faces. To solve the false rejection problem, we use another definition of DFFS+DIFS presented in [39]. Instead of fitting a Gaussian to coefficients in each dimension, the distance of the worst outliers in each dimension are selected as the σ value for the

multi-variance Gaussian. The i^{th} σ value σ_i can be calculated as:

$$\sigma_i = (\sum_{j=0}^{j<N-1} (\lambda_{j_i})^{\infty})^{-\infty},$$

where N is the number of the training images. Sequentially, the σ value for residual error equals the distance of the worst outliers:

$$\sigma_{residue} = (\sum_{j=0}^{j<N-1} (\varepsilon_j)^{\infty})^{-\infty}.$$

The probability distribution function of the "faceness" value can then be formulated as:

$$faceness(\lambda_0, \ldots, \lambda_{M-1}, \varepsilon) = \prod_{k=0}^{M-1} \exp(-\frac{\lambda_k^2}{2(\sigma_k)^2}) \exp(-\frac{\varepsilon^2}{2(\sigma_{residue})^2}) \tag{3.9}$$

The new definition of DFFS+DIFS can be obtained by computing the logarithm of Equation 3.9:

$$DFFS + DIFS(\lambda_0, \ldots, \lambda_{M-1}, \varepsilon) = k \times (\sum_{k=0}^{M-1} \{\frac{\lambda_k^2}{\sigma_k^2}\} + \frac{\varepsilon^2}{\sigma_{residue}^2}). \tag{3.10}$$

Note that k is an arbitrary constant used as a scale factor. The first term in the parenthesis is the corresponding *distance in feature space* value (compare to Eq. 3.8), and the second term is the *distance to feature space* value (compare to Eq. 3.7).

The DFFS+DIFS value can be used to assess the quality of the AAM fitting result since it yields low value for good face model fitting and high value for poor fitting. The DFFS+DIFS value is not exactly zero for perfect fitting result since only the mean face is located precisely in the center of the cloud representing the distribution. All the other face instances have a distance from the center of the cloud thus have a non-zero DFFS+DIFS value.

With this distance measure, the quality of the AAM fitting can be quantified as an indicator of the fitting accuracy. The AAM tracking, actually, may easily lose the target because of large motion, extreme head rotation and occlusion. A large DFFS+DIFS value indicates that the tracking/fitting fails and new initialization is required to continue tracking on the following frames.

As mentioned above, the main factors that cause the AAM fitting to lose the target across a video sequence are large motion, extreme head rotation and occlusion. A target face in a scene may move very fast and the appearance of the face will become blurred if the recording frame-rate is not high enough. The

blur effect makes it difficult for the AAM fitting on the one hand and on the other hand, the face detector fails due to the blur effect as well. In such a case, the re-initialization is impossible. Fortunately, there are other cues to localize a face. The skin color model is able to segment the face region from the background. However, the pixels in the background region may also contain pixels in skin colors, which makes the face localization difficult. To tackle this problem, we employed a robust face tracker based on particle filters [27, 38]. The particle filter tracks the face in a scene by propagating the conditional distribution of the tracking target frame-by-frame. The sampled particles are considered as the approximation of the conditional distribution. In [27], the score of the Adaboost classifier [64] and the proportion of the skin color pixels are used as the confidence of a particle. For technical details of the face tracker please refer to [27]. Here in this work, we utilize the particle filter-based face tracker to track the face, while a minor change is made: instead of using a fixed skin color histogram in HSV color space, we update the skin color histogram according to the previous successful fitting results. The pixels inside the fitted mesh are used to update the old skin color histogram. In addition to the skin color histogram, a non-skin color histogram is generated from the background pixels. The decision for skin color detection is made by employing the Bayesian classifier [20]:

$$\frac{p(x|skin)}{p(x|nonskin)} \geq \tau, \tag{3.11}$$

where $p(x|skin)$ and $p(x|nonskin)$ are the respective class-conditional probability density functions (pdfs) of skin and non-skin colors and τ is a threshold. The theoretical value of τ that minimize the total classification cost depends on the a priori probabilities of skin and non-skin and various classification costs; however, in practice τ is determined empirically. The class-conditional pdfs are estimated using the skin and non-skin color histogram respectively.

However, the particle filter-based face tracker is not able to track the face in case of large angle rotate (pitch, yaw and roll), because the confidence values of the particles obtained from the Adaboost classifier are too low to keep on tracking the target. If the face does not move too fast, the AAM face tracker is able to track the face even it rotates in plane or in depth, since faces with different rotation angles ($\leq 60\,^{\circ}$) are modeled in the AAM. Shape parameters for global shape normalization are also optimized during the fitting iterations.

3.4 Pose normalization

The main goal of this thesis is to use the AAM face fitting method to improve face registration. After the face fitting, we describe how a fitted face is normalized with various pose normalization methods.

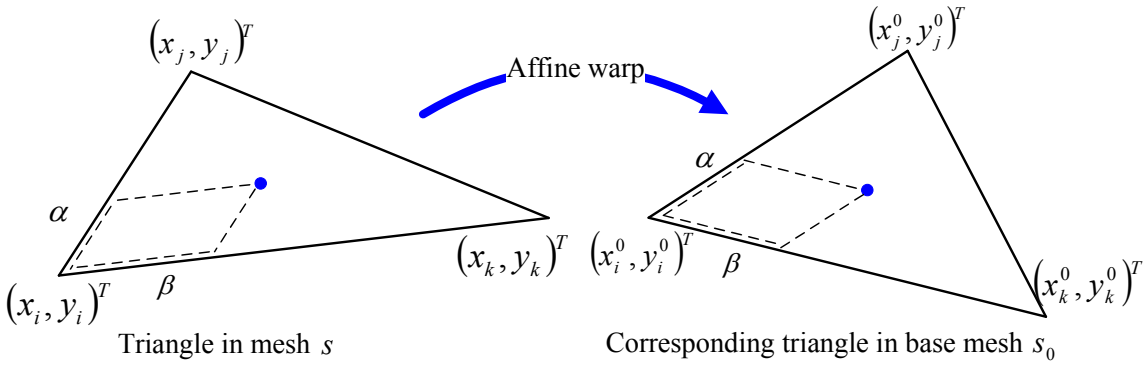


Figure 3.7: Back projection with piecewise affine warp.

3.4.1 Piecewise affine warping

The most straightforward method is the piecewise affine warping which is also used in the fitting iteration for sampling texture inside the face mesh. The warp is realized by mapping the pixels in the fitted triangular mesh s to the base mesh s_0. For each pixel $\mathbf{x} = (x, y)^T$ in a triangle in the base mesh s_0, it can find a unique pixel $\mathbf{W}(\mathbf{x}; \mathbf{p}) = \mathbf{x}' = (x', y')^T$ in the corresponding triangle in the mesh s by an affine mapping. One way to implement the piecewise affine warp is depicted in Figure 3.7. Consider the pixel $\mathbf{x} = (x, y)^T$ in the triangle $(x_i^0, y_i^0)^T$, $(x_j^0, y_j^0)^T$, and $(x_k^0, y_k^0)^T$ in the base mesh s_0. This pixel can be uniquely expressed as:

$$\mathbf{x} = (x, y)^T = (x_i^0, y_i^0)^T + \alpha[(x_j^0, y_j^0)^T - (x_i^0, y_i^0)] + \beta[(x_k^0, y_k^0)^T - (x_i^0, y_i^0)] \quad (3.12)$$

where:

$$\alpha = \frac{(x - x_i^0)(y_k^0 - y_i^0) - (y - y_i^0)(x_k^0 - x_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (y_j^0 - y_i^0)(x_k^0 - x_i^0)}$$

and:

$$\beta = \frac{(y - y_i^0)(x_j^0 - x_i^0) - (x - x_i^0)(y_j^0 - y_i^0)}{(x_j^0 - x_i^0)(y_k^0 - y_i^0) - (y_j^0 - y_i^0)(x_k^0 - x_i^0)}$$

The warped point $\mathbf{W}(\mathbf{x}; \mathbf{p})$ in the corresponding triangle in s is then:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = (x_i, y_i)^T + \alpha[(x_j, y_j)^T - (x_i, y_i)^T] + \beta[(x_k, y_k)^T - (x_i, y_i)^T]$$

where $(x_i, y_i)^T$, $(x_j, y_j)^T$, and $(x_k, y_k)^T$ are the vertices of the corresponding triangle in s.

The piecewise affine warping is simple, and the deformation field is not smooth. This is reflected in Figure 3.8. Straight lines may usually be bended across triangle boundaries.

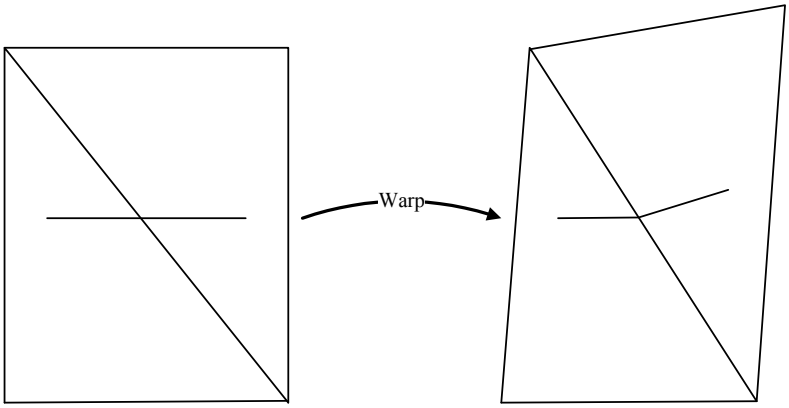


Figure 3.8: Problem of the piecewise affine warping. Straight lines will usually be bended across triangle boundaries.

3.4.2 Thin-plate spline warping

To overcome the discontinuity of piecewise affine warping across the triangle boundary, we investigated another warping technique based on thin-plate spline (TPS) [8] to improve the smoothness of deformation. However, the behavior of the method is not always clear in-between feature points, especially in the case of large pose variations.

3.5 Face recognition

After the faces are registered and pose normalized, face recognition takes place.

3.5.1 Feature extraction & selection

The synthesized frontal view of faces are masked with a binary mask image. All important facial features, including eyes, nose, and mouth, are warped into the same coordinate framework. However, feature points around the chin area might be misaligned which may affect the recognition performance. As demonstrated in [25], however, the chin area does not contain too much discriminative information compared to other facial features. For this reason, we decide to crop the chin area in the synthesized frontal face image. The cropped image is then scaled to 64×64 pixels size.

The cropped and scaled images can be considered as a 64×64 dimensional vector. Using all the raw gray-level value to perform classification is neither efficient nor effective. Therefore, the dimensionality of the feature vector has to

be reduced. Following the method described in the basic principles part, a scaled face image is divided into 64 blocks of 8×8 pixels size. On each 8×8 pixels block, a DCT is performed. The obtained DCT coefficients are ordered using zig-zag scanning as shown in Figure 2.10. The first component in the obtained DCT coefficients is called DC, which will be skipped because it represents the average pixel intensity of the entire block. The following ten low frequency coefficients are retained and the remaining coefficients are discarded. This process yields a ten dimensional local feature vector. Finally, the 64 local feature vectors are concatenated to construct the feature vector of a whole face image.

DCT preserves the total image energy of the processed input block, therefore blocks with different brightness levels lead to DCT coefficient with different magnitude values. In order to balance each local block's contribution to the classification the local feature vector is normalized to unit norm. The magnitude of feature vector f is transformed into the normalized feature vector f^n:

$$f^n = \frac{f}{\|f\|}.$$

Another normalization method [23] is also considered, which divides each of the DCT coefficient with its standard deviation:

$$f^C_{i,j} = \frac{f_{i,j}}{\sigma(f_j)},$$

where $f_{i,j}$ is the j^{th} DCT coefficient form the i^{th} block in an image, $\sigma(f_j)$ is the standard deviation of the j^{th} DCT coefficient. The combined normalization is done by normalizing f^C_i to unit norm:

$$f^{n,C}_i = \frac{f^C_i}{\|f^C_i\|}.$$

3.5.2 Classification

Since we considered face registration both in still images and in video sequences, face recognition based on this registration approach is both carried out on still image data set as well as video sequences.

For still image face recognition, we use the nearest neighbor classifier with $L1$ norm as the distance metric. It has been shown that the $L1$ norm provides better results than the L2 norm and normalized correlation. A test sample face is assigned to the class of the closest training sample face.

Face recognition across video sequences using the proposed face alignment method is also studied. The experimental dataset was originally collected for a study of open set face recognition [61]. We also adopted the classification

methodologies from it. The support vector machines (SVMs) [63] are employed as the classifiers for open set recognition. Comparisons of performance between SVMs and nearest neighbor were also made for open set recognition and the results achieved by using the SVMs outperform those achieved by using nearest neighbor.

The detailed methodology is stated in [61]. For ease of explanation, we only give the performance criterion for open set recognition. For other details, please refer [61].

Open set recognition can be considered as a multi-class verification problem. When classifying a known person in open set recognition, the system can either correctly accept the person as known and identify him or her correctly, correctly accept but falsely classify the person, or the person may be falsely rejected. An unknown person can be correctly rejected or falsely accepted. Taking these into account, the performance measures are defined as: Correct classification rate (CCR) (correct acceptance and identification of known), correct rejection rate (CRR) (correct rejection of unknown), face acceptance rate (FAR) (false acceptance of unknown), face rejection rate (FRR) (false rejection of known) and false correction rate (FCR) (correct acceptance but misclassification of known).

The error rates are defined as

$$FAR = \frac{n_{impostor,accepted}}{n_{impostor}}$$

$$FCR = \frac{n_{genuine,misclassified}}{n_{genuine}}$$

$$FRR = \frac{n_{genuine,rejected}}{n_{genuine}}$$

where $n_{genuine}$ and $n_{impostor}$ represents the number of genuine and impostor samples presented to the recognition system.

4 Experiments

This chapter will present the dataset and the experiments that were conducted on the data. We first start by describing the data set used in this thesis. Afterwards, still image face recognition based on the proposed face registration method is analyzed. The fully automatic facial feature localization experiments are presented in Section 4.3.2. Finally, Section 4.3.3 gives details about the performance of the open set face recognition in videos with the presented face tracking and alignment technique.

4.1 Experiment setup

Three major experiments were conducted on different data sets in this study. The first experiment was designed for the evaluation of the face alignment with pose variations. The $b*$ subset of the FERET [55] database was chosen, which contains at most $\pm 60^\circ$ of face rotation in depth. The second experiment was conducted on the near frontal face images in FERET and the BioID data set [1] to evaluated the performance of facial feature localization using the model fitting algorithm. The third experiment was about the open set face recognition in video sequences, which was performed on a data set that was recorded by us between January and May 2007. The data set consists of a real-world data collected in a wide hallway in front of an office.

The AAM model used for all experiments was trained as a generic model. The model training process is described in Section 3.1. After data refitting process, the resulting AAM has 15 shape bases and 60 appearance bases, which account for 95% of the shape and appearance variations in the training set. The mean shape has 92×98 pixels resolution. The appearance of the AAM is modeled in gray-level texture, since some of the evaluation images are gray-level images and the face recognition approach is also based on pixel intensity.

The enhanced DCT feature extractor was used in the first experiment, in which local DCT coefficients are first divided by the standard deviations of the corresponding frequency coefficients, and transformed to unit norm subsequently [23].

The experiment on open set recognition is based on the work [61] by Szasz-Toth. The implementation for SVM-based classification is based on LibSVM [11].

SVM parameters	
SVM type	C-SVM classification
SVM kernel	Polynomial $(\gamma x_i x_j + coef0)^d$
Polynomial degree d	2
γ	2
C	32
$coef0$	0
ϵ	10^{-10}

Table 4.1: SVM parameters

The SVM parameters were optimized by performing a grid search on an independent cross-validation set based on the FRGC data. The parameter values are listed in Table 4.1

4.2 Experimental data

We conducted three different experiments on different data sets. The following subsections describe the data sets used for different evaluation purposes.

4.2.1 Data set 1

The first data set is a subset of the Facial Recognition Technology (FERET) database [55]. This subset contains 200 subjects, where each subject has 11 images. All images of an example subject are shown in Figure 4.1. The label ba indicates the frontal series, which was used as the training set (gallery) in this experiment. Series bj contains alternative expression and bk was recorded under different illumination condition. Both bj and bk are frontal face images. The remaining eight series are non-frontal face images with different pose angles ($\pm 60\,^\circ$, $\pm 40\,^\circ$, $\pm 25\,^\circ$, and $\pm 15\,^\circ$). Positive pose angle means that the subject faces to his left and negative pose angle means that the subject faces to his right. All except the ba series were used for testing (probe). None of the subjects used from the FERET for training the AAM were used during the recognition experiments.

4.2.2 Data set 2

The second data set was selected for the facial feature localization experiment. Here, we used the frontal face subset in the FERET database (FERET-frontal) and

Figure 4.1: Example images from the FERET $b*$ series. ba indicates frontal face; bj with expression variation; bk with illumination variation; $b[b-i]$ with different pose angles: bb (60 °), bd (40 °), bc (25 °), be (15 °), bf (−15 °), bg (−25 °), bh (−40 °), and bi (−60 °).

BioID [1] for this purpose. FERET-frontal is a publicly available database with 3880 images which were taken under controlled settings without background clutter and significant variations in illumination. The publicly available BioID database contains 1521 frontal face images that contain variations with respect to illumination, background, face size and head pose. The recording condition is less controlled compared to the FERET-frontal database which makes the feature localization task more challenging.

4.2.3 Data set 3

To evaluate the performance of AAM face tracking in video sequences, the video data used for open set face recognition was utilized. 55 people were recorded in front of an office. Lighting was natural or artificial, depending on the time of the day. These recordings were split into two groups, a group of known people and a group of unknown people.

Different sets of data were used for training and testing. Known people's recordings were split into training and testing sessions which do not overlap. Unknown subjects used for training are different from those used for testing.

Figure 4.2 depicts some example frames in the open set database. Since it was recorded with a normal web-cam and the frame rate is low (≈ 10 fps), the face blurred when it moves to fast(Figure 4.2(a)). The recorded subjects were free to move, even out the sight of the camera. The depicted example frames show different recording conditions as well as subjects with various pose.

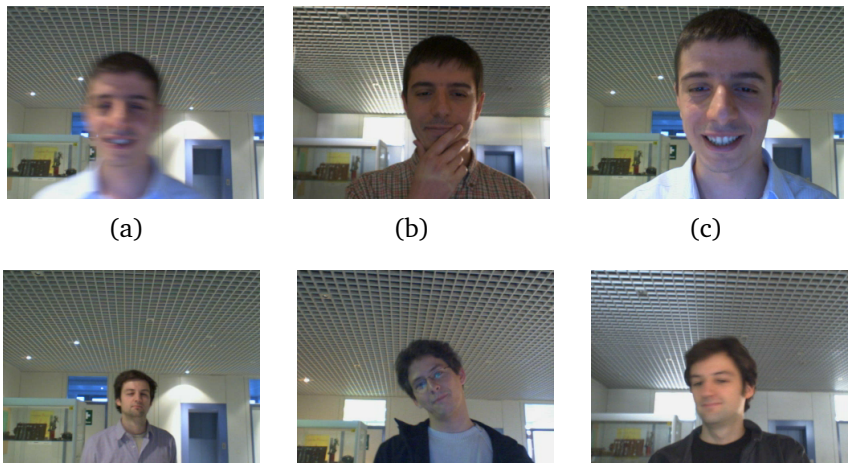
4.3 Experimental results

This section presents the experiments in detail as well as the performance analysis.

4.3.1 Results of face recognition on still images

The goal of this experiment is to assess the contribution of AAM-based face registration for the local appearance-based face recognition. The experiment was carried out on the FERET $b*$ subset which contains large pose variations. With the AAM fitting approach, we generated a pose normalized frontal-view face for each input image, and face recognition was based on the frontal-view face image.

The FERET ba series, which only contains near frontal face images, was selected as the training set. The remaining series (bb-bk) were considered as different testing sets, which means face recognition was evaluated on different series separately.



(a) (b) (c)

(d) (e) (f)

Figure 4.2: Recordings from the data set, different illumination and face sizes. (a) Artificial light, motion blur. (b) Day light, dark, partial occluded. (c) Artificial light, bright, near. (d) Artificial light, far away. (e) Head rotate in plane. (f) Head rotate in depth.

The corresponding synthesized frontal-views of the face images in Figure 4.1 are displayed in Figure 4.3. The automatic AAM initialization may not be always stable because of the large angle rotation of the face in some series such as bb and bi. The Haar-cascade-based facial feature detectors fail to detect eyes and mouth, which are important for a precise AAM initialization. In this sense, we used a semi-automatic AAM initialization where the annotation of eye-centers and mouth-centers were used for model initialization. The model was fitted progressively, i.e. first fit an inner-face AAM, and then expand the inner-face model to a whole-face model and process the second fitting.

In general, the fitting performance on the near-frontal faces is better than on the semi-profile faces. While fitting semi-profile faces, even a small misalignment in the chin area may cause large error in the synthesized face image. The reason for this problem is that the partially self-occluded face part is over-sampled during the AAM warping. The misalignment error is enlarged through this over-sampling. Another fact that we have noticed is that even a semi-profile face is fitted perfectly, the synthesized frontal-view face still looks different from the real frontal face. Take the bi in Figure 4.3 for an example, the left face is over-sampled and the right face is down-sampled after AAM warping. This effect makes the left-eye wider and the right-eye narrower, which results in different local DCT frequencies compared to the training image ba in Figure 4.3.

We considered two possible pose normalization approaches in section 3.4. We

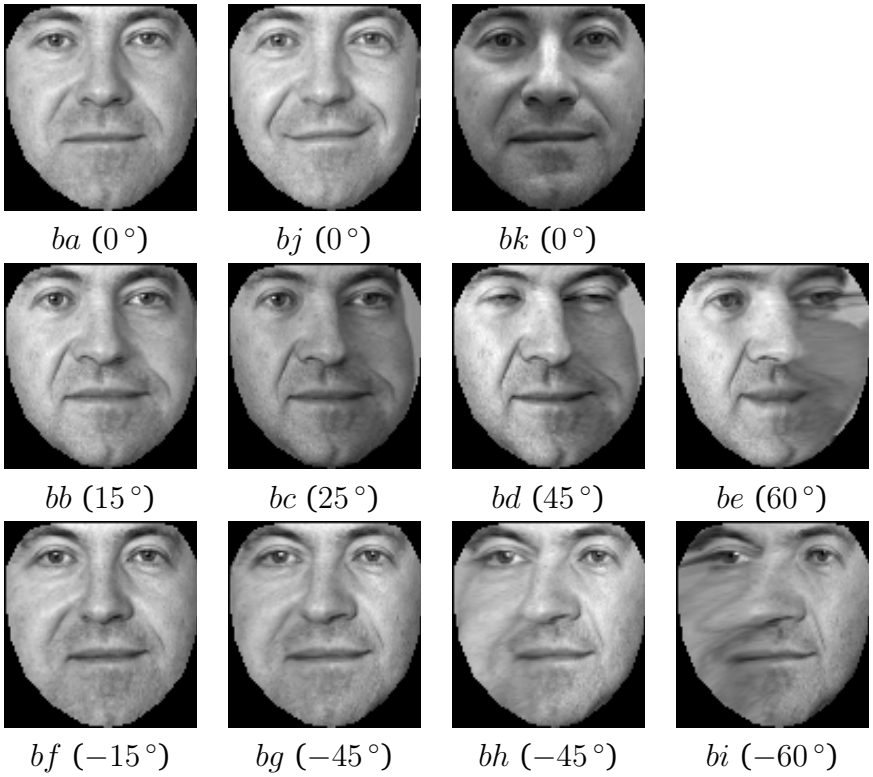


Figure 4.3: Fitted faces with pose normalization (Corresponding to Figure 4.1).

started the first experiment with different pose normalization and face synthesis methods. First, the piecewise affine warp was considered. The original piecewise (PA) warp stretches zero-mean unit-length texture vector defined inside the base mesh s_0 to $[0-255]$ intensity values. In addition to PA, piecewise affine warping with original pixel intensities was also considered and we named it PAN (Piecewise Affine warp No stretch). The third synthesis method (HIST) was also based on the piecewise affine warp, however, the histogram of the synthesized face was matched to the histogram of the mean face. In addition to the piecewise affine-based warping methods, the TPS-based warping technique was also evaluated. The face recognition results based on these four face synthesis approaches are plotted in Figure 4.4. Note that the recognition performances of these four face synthesis methods are similar when testing on be and bf, where the face rotation degrees are at most $\pm 15\,^{\circ}$. With the increasing degree of face rotation in depth, the recognition results differ from each other. Face synthesis with PAN outperforms the others in most cases. The performance of the other piecewise affine-based methods decreases dramatically when the fitting performance decreases due to large degree of face rotation. The reason for this behavior is that both methods change the histogram of the original input face image. In case of misalignment of the outline shape points, the background pixels may effect the histogram in original face area. In extreme case, if the background intensity is very light, after intensity stretch the pixels in face area become very dark. The TPS-based method, however, does not help in terms of face recognition. It achieves the worst results when testing on bi, bh and bf.

In addition to the different face synthesis methods, we also compared various face recognition techniques to evaluate how well the local appearance-based face recognition performs on the proposed face registration and synthesis methods.

Since the training set only contains one example per subject, the selection for other face recognition techniques is limited. Techniques which utilize intra-class information are not suitable for this evaluation. We selected two well-known face recognition approaches: Eigenfaces (PCA) [62] and embedded hidden Markov models (EHMM) [50].

We performed the selected algorithms on the synthesized face images with the PA method. The evaluation was also carried out on the same data sets as in the last experiment. Figure 4.5 plots the corresponding recognition rates. DCT denotes our local appearance-based face recognition approach since it use the local DCT coefficients as the feature vector. For the Eigenfaces approach, 40 eigen-components were used which yielded optimal performance.

We noticed that DCT outperformed the other two holistic approaches significantly over all the eight series. Moreover, the performances of EHMM and PCA decreased drastically when the rotation angles increased. This demonstrates again that the performance of the holistic approaches can be easily effected by

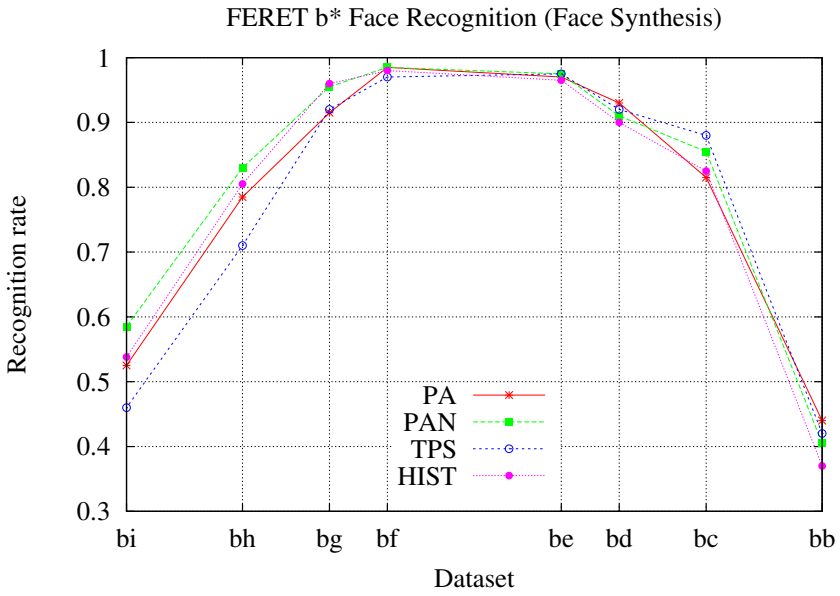


Figure 4.4: Face recognition on the FERET $b*$ series with different face synthesis techniques.

local appearance. Here, the texture on the self-occluded face part is inconsistent to the corresponding part in the training image.

As we observed, the performance of the face fitting decreases as the face rotation angle increases. Usually the misalignment occurs in the chin and cheek area where the left or right part of the face is self-occluded by the right or left face. The half face which is not occluded can be fitted as usual. An intuitive idea can be: why don't we perform face recognition on the well fitted half faces? We conducted this experiment again using DCT-based face recognition. However, we only extract DCT features on the left half faces when the face rotation angles are positive (subset bb, bc, bd and be) and on the right half faces when the face rotation angles (subset bf, bg, bh and bi) are negative. The results of this evaluation is depicted in Figure 4.6. The recognition rate using half faces did not increase in comparison with the full face recognition except on the subset bc. The results indicate that, even if the half face is fitted without misalignment, the down-sampling on that half face still affects the performance of recognition. And somehow the other part of the face contributes some discriminating information, although it may be misaligned or over-sampled. In [36], face recognition using half face was also investigated. Unfortunately, they also reported that the full face recognition outperformed the half face recognition.

We end this section by comparing the proposed face fitting and alignment

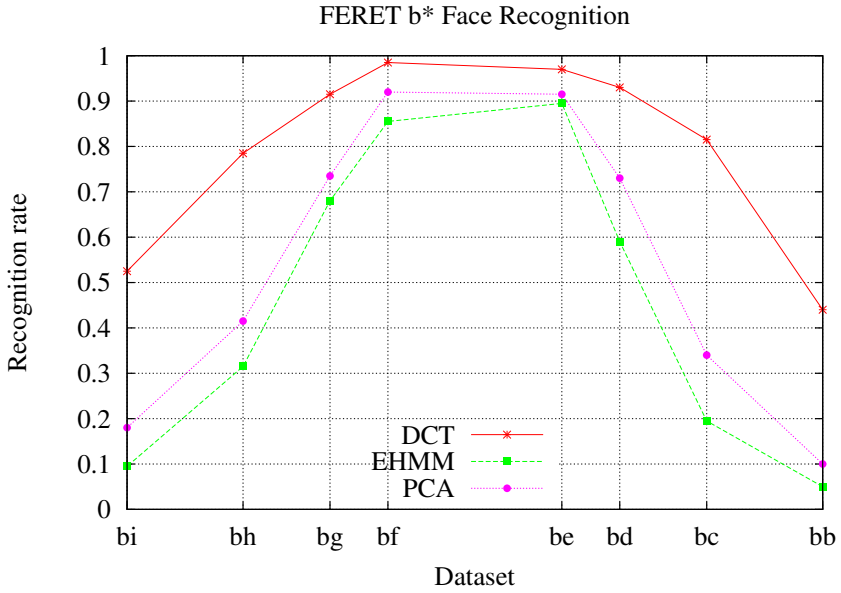


Figure 4.5: Comparative analysis of face recognition methods on the FERET $b*$ series.

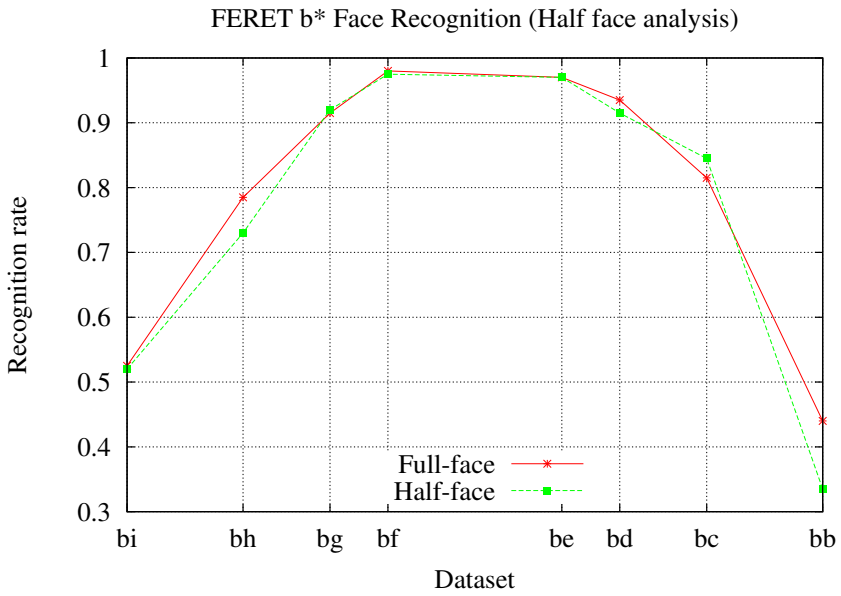


Figure 4.6: Face recognition on the FERET $b*$ series (Full face vs. half face).

Probe set	bb	bc	bd	be	bf	bg	bh	bi
Rec. rate (PA)	44.0%	81.5%	93.0%	97.0%	98.5%	91.5%	78.5%	52.5%
Rec. rate (SA)	0.0%	5.5%	26.0%	62.5%	78.5%	26.5%	4.0%	1.0%

Table 4.2: Face recognition results on the FERET subset bb-bi. Face synthesis with piecewise affine (PA) warp and simple affine (SA) warp. Recognition with enhanced DCT feature.

Probe set	bj	bk
Rec. rate (PA)	82.5%	97.0%
Rec. rate (SA)	86.0%	76.5%

Table 4.3: Face recognition results on the FERET subset bj and bk. Face synthesis with piecewise affine (PA) warp and simple affine (SA) warp. Recognition with enhanced DCT feature.

method to the simple affine face alignment. It should be noted that, it is not fair to make a comparison between these two methods since the simple affine method does not correct face pose. The results demonstrate that the face recognition performance increases significantly with the pose correction. As listed in Table 4.2, recognition with simple affine alignment only achieved 62.5% and 78.5% with $\pm 15\,^\circ$ degree of face rotation. And it did not work at all when the face rotation angles ware larger than $40\,^\circ$. Table 4.3 listed the results evaluated on the other two series: bj and bk. Both series contain near frontal faces, however, face images in bj vary in expression while bk differs in illumination from ba. This time, the results with simple affine were better than AAM-based alignment on the series bj. This implies that the proposed face alignment approach is not suitable for data sets with expression variation due to deformation.

4.3.2 Results of feature localization

After analyzing the effectiveness of the proposed face alignment in terms of face recognition, we need to evaluate the performance of the automatic face alignment for building a fully automatic face recognition system.

The eye-centers are important facial feature and both the AMM initialization and the simple affine face alignment are based on the location of the eye-centers. We evaluated eye localization on the FERET-frontal and BioID database. As we stated earlier, since it is easier to fit an inner-face AAM than a whole-face AAM, we only fit the inner-face AAM to the images in the databases. The performance of left-eye and right-eye localization is evaluated separately. Figure 4.7 plots the localization results on the FERET-frontal with respect to inter-ocular

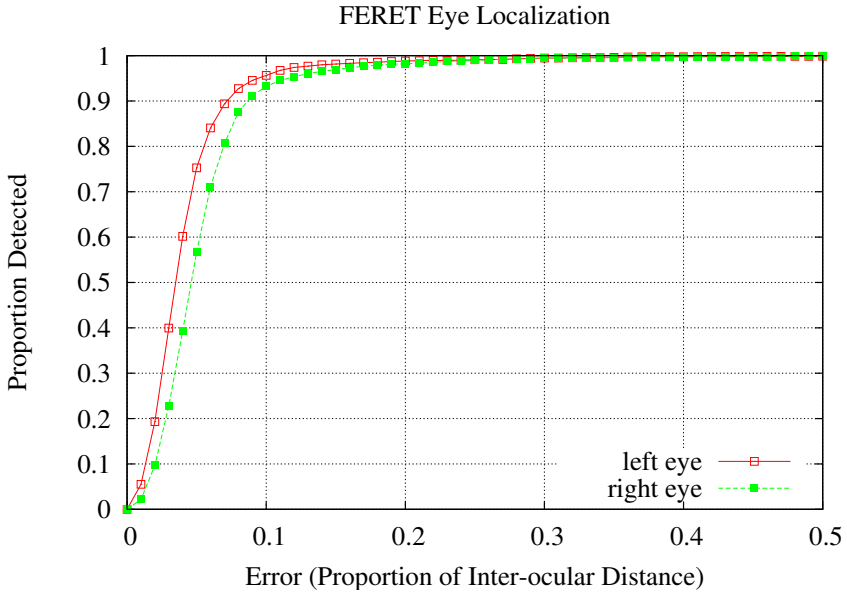


Figure 4.7: Performance of eye localization on the FERET-frontal data with respect to inter-ocular distance.

distance error. Although only the shape and appearance of the inner face are fitted, the underlying idea is still a holistic template matching method. The precision of the eye localization can be affected by mismatch of other facial features such as nose and mouth. Compared to the other feature-based eye localization methods [26], the results are median. The eye localization results on the BioID database is plotted in Fig 4.8. As the BioID database were collected under a more uncontrolled condition, the localization results were worse than the ones obtained on the FERET-frontal.

As we intended to use the DFFS+DIFS distance metric to indicate the quality of a fitting (for ease of notation, we denote DFFS+DIFS as DFFS hereafter), we need to study the relationship between the DFFS and the shape matching error. In this section, the relationship between the DFFS and the eye localization error is plotted in Figure 4.9 and Figure 4.10 for the FERET-Frontal and BioID databases respectively. The plots show that high inter-ocular distance error always implies a high DFFS value, which means poor eye localization results in high DFFS value. But the inverse is not correct. Sometimes the eyes were localized with a low inter-ocular distance error, but the DFFS value is rather high, which means that the other part of the inner face does not match at all. For AAM-based face registration, the registration fails since not all of the model shape points are correctly placed on the corresponding feature point.

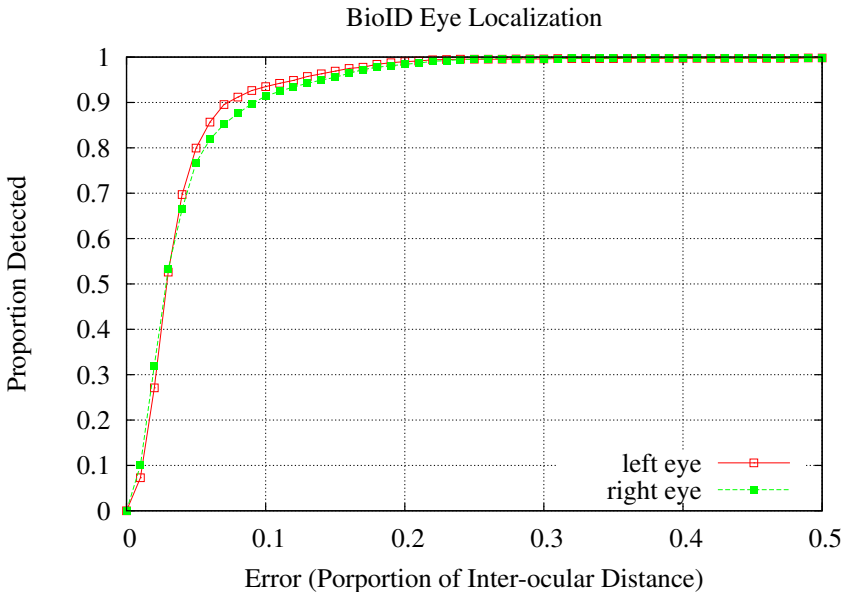


Figure 4.8: Performance of eye localization on the BioID database with respect to inter-ocular distance.

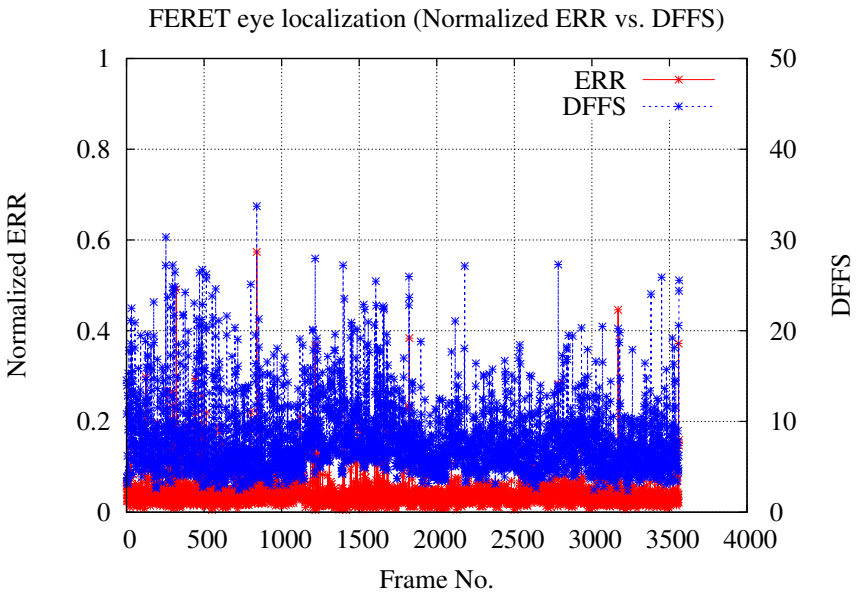


Figure 4.9: Performance of eye localization on the FERET-frontal data (inter-ocular error distance vs. DFFS).

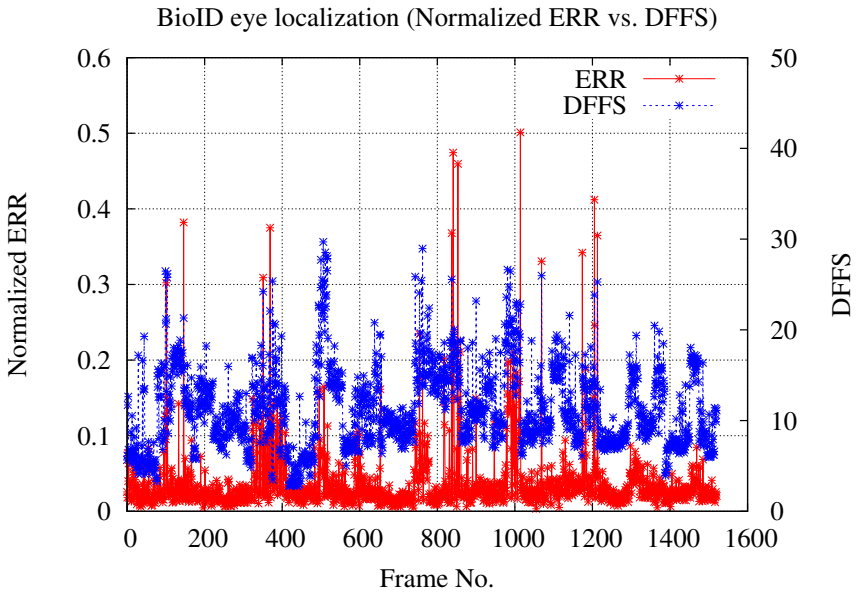


Figure 4.10: Performance of eye localization on the BioID database (inter-ocular error distance vs. DFFS).

4.3.3 Results of face recognition on video sequences

As we know, AAM fitting on still images is difficult because of initialization. Usually, the AAM-based face alignment is investigated for face tracking in video sequences. With proper model initialization, fitting faces on continuous frames is more feasible.

4.3.3.1 Performance evaluation on AAM face tracking

We investigated AAM tracking on the open set video data set. To evaluate the tracking performance, we annotated a representative video with 178 frames. Motion blur, rotation in plane and depth were recorded in this video, which simulates a real world scene. The tracking was initialized automatically based on the Haar-cascade face detector and feature detector. Figure 4.11 plots the tracking error in terms of pixel-wise landmark location error and the DFFS value. This plot indicates a clear correlation between the fitting error and the DFFS value. High DFFS value implies large fitting error.

Figure 4.12 depicts the tracking resulted with manual initialization in the first frame. Comparing it with Figure 4.11, the manual initialization does result in lower shape fitting error in the first few frames. However, the fitting performance in the following frames was similar to the automatic initialization. In

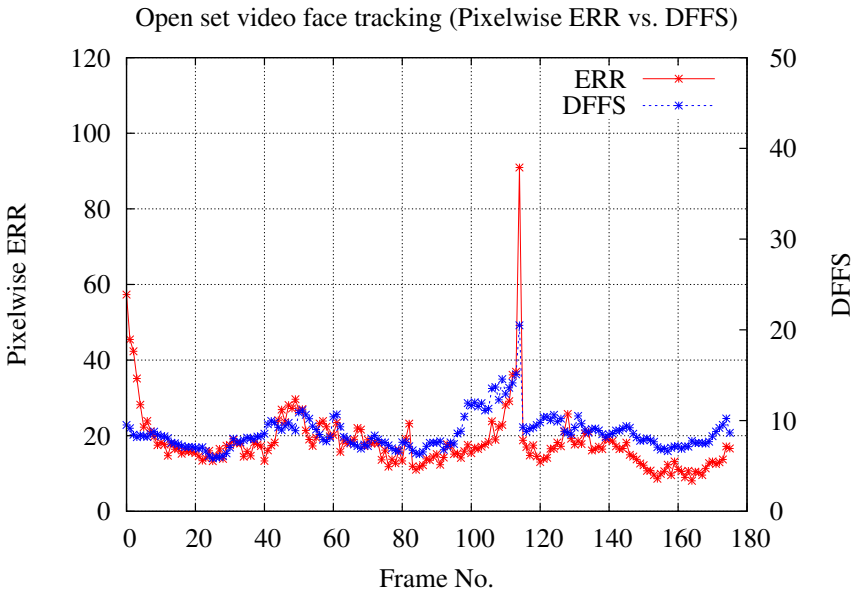


Figure 4.11: AAM face tracking with SICOV and automatic initialization (pixelwise shape error vs. DFFS).

both tracking processes, the model re-initialization was fully automatic which was based on the face detector, feature detector and particle filter-based face tracker.

Fig 4.13 compares the tracking performance with the SIC fitting algorithm and the SICOV algorithm. The SICOV fitting algorithm not only matches the model appearance, but it also considers the previous fitted appearance in the video sequence. We observed that, with the SICOV algorithm, the tracking was more stable than only fitting the model appearance. The subject's face rotated in plane with large angle near the frame number $100 - 120$. The fitting was successful when tracking with the SICOV algorithm while the SIC missed the target and yielded large shape fitting error.

After analyzing the AAM tracking in video sequences, we carried out open set face recognition based on the proposed AAM face tracking and alignment approach. The tracking was based on the SICOV algorithm with automatic model initialization. The illumination effect on the fitting was remedied by fitting the histogram of the warped face image to the histogram of the mean face. After fitting each frame, the frontal-view of the fitted face is synthesized with the PAN (Piecewise Affine warp No stretch) approach.

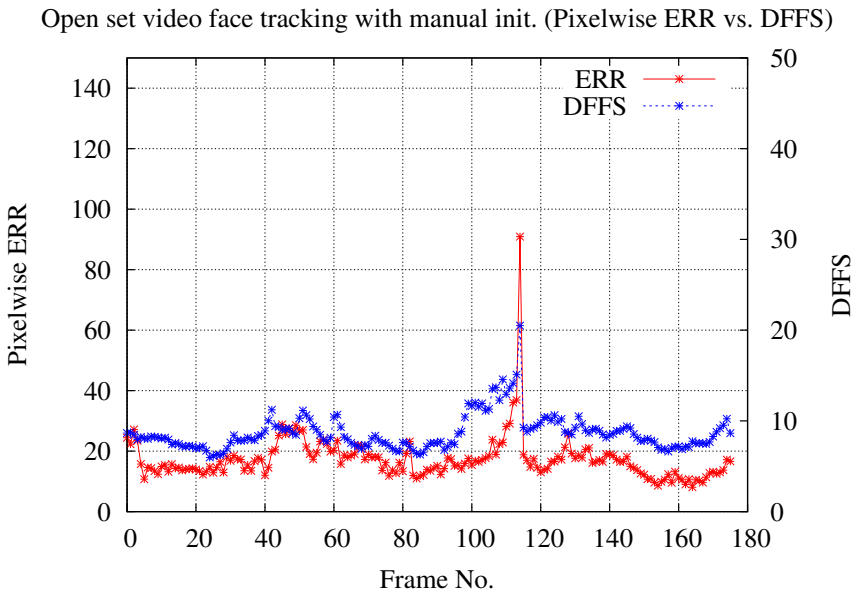


Figure 4.12: AAM face tracking with SICOV and manual initialization (pixelwise shape error vs. DFFS).

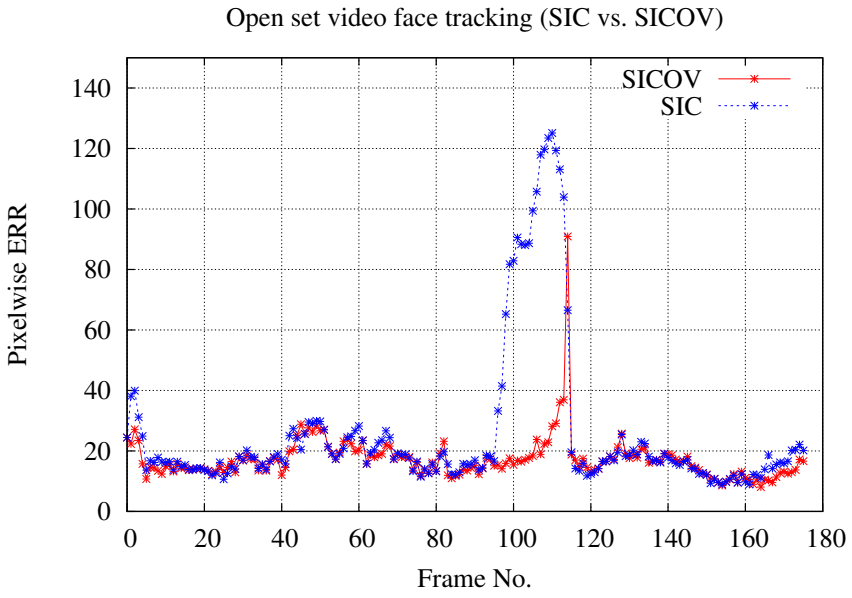


Figure 4.13: AAM face tracking (SIC vs. SICOV)

4.3.3.2 Performance measure for open set face recognition

Before conducting the open set face recognition experiment, we explain some performance measure for open set recognition.

Different from close-set recognition, we have to minimize three error terms as defined in Section 3.5.2, namely FAR, FRR and FCR. However, it's not possible to minimize the three error terms at the same time, therefore the equal error rate (EER) performance measure is employed to trade off against the three error terms.

The EER can be found by choosing a threshold for which

$$FAR = FRR + FCR. \tag{4.1}$$

Support vector machines automatically minimize the overall error and try to find the global minimum. Therefore if the decision hyperplane is not altered these errors are all automatically minimized. To fine tune the system performance, the ROC curve for SVM-based classification was created by using a parameterized decision surface [53]. The decision hyperplane $\{x \in S : wx + b = 0, (w, b) \in S \times R\}$ is modified to $wx + b = \Delta$, where Δ allows to adjust the false acceptance rate and the correct classification rate accordingly.

4.3.3.3 Performance comparison

Table 4.4 lists the data configuration for the training and testing. As suggested in [61], the unknown training data was under-sampled so that the number for images from unknown subjects and each known subject is balanced. Note that only the frames fitted under a certain DFFS threshold were accepted for training and testing. The threshold was selected empirically so that it discards all possible misalignment while keeping outliers. Here we choose $DFFS_{threshold} = 10.0$. After face registration with proposed method through the known training sequences, we obtained approximately 600 known training samples for each subject. 25 subjects with single session were used for unknown training, the frames were under-sampled to 30 frames per subject.

We first started with frame-based classification where the face registration was based on the AAM fitting and piecewise affine warping. The results are listed in the first line of Table 4.5 which were calculated using $\Delta = -0.23$ for the SVM parametric hyperplane, where $EER = 4.4\%$. Receiver operating characteristic (ROC) curves for this test are plotted in Figure 4.14. The ROC curves plot the correct classification rate against the percentage of impostors accepted by the system. The x-axis represents the FAR and the CCR is plotted on the y-axis.

Another frame-based test was also carried out on the same data set to verify the effectiveness of the pose correction. Instead of synthesizing face with piecewise affine warp after AAM fitting, a simple affine warp was used according to

Training data		
Known	5 subjects	4 sessions and ≈ 600 frames per person
Unknown	25 subjects	1 session, 30 frames per subject
Testing data		
Known	5 subjects	3-7 sessions per person
Unknown	20 subjects	1 session per person

Table 4.4: Data set for open set experiments

the eye coordinates in the fitted AAM shape. This method resizes and crops the face image in a rectangle, and the eye centers are constrained in a fixed location in the face rectangle. The corresponding frame-based recognition is listed in the first line of Table 4.6 where $EER = 3.8\%$ and $\Delta = -0.16$. And the ROC curve for this test is depicted in Figure 4.15.

The correct recognition rates in both tests were high. However, recognition using simple affine warp-based registration slightly outperformed the piecewise affine warp based registration. The reason for this result is that although piecewise affine warp-based face registration is able to correct pose, the frames with large angle face rotations were usually discarded due to the imprecision of fitting compared to near frontal faces, as well as the face synthesis problem which is demonstrated in section 4.3.1. Furthermore, in the case of facial expression variations, the recognition results based on piecewise affine warp degenerated significantly as observed in Table 4.3.

The frame-based face recognition in video sequences makes decision on every single frame. The results, therefore, return some insight on the general performance of the registration and classification scheme employed. On the other hand, this simple approach does not utilize any additional information contained in video sequences. Hence in addition to frame-based classification, we investigated other two classification schema for face recognition in video sequences.

The first considered approach is progressive-score-based classification. Instead of classifying frames independently, a simple temporal fusion is applied by accumulating frame scores over time. This can be thought of as classifying every frame as if it were the end of a sequence and taking the final score. The second approach is much like the progressive-score-based classification, however, the decision is returned only at the end of the entire sequence. Thus, it is referred as video-based classification. The results with progressive-score-based and video-based classification are listed in Table 4.5 and Table 4.6, respectively for two registration approaches.

Observing the results based on the progressive-score-based classification, we noticed that the results were improved compared to the frame-based scheme.

Classification	CCR	FRR	FAR	CRR	FCR
Frame-based	95.9%	3.9%	4.4%	95.6%	0.2%
Progressive-score	98.1%	1.9%	2.7%	97.3%	0.0%
Video-based	100.0%	0.0%	0.0%	100.0%	0.0%

Table 4.5: Classification results with AAM face synthesis

Classification	CCR	FRR	FAR	CRR	FCR
Frame-based	96.4%	3.5%	3.8%	96.2%	0.1%
Progressive-score	97.1%	2.8%	3.0%	97.0%	0.1%
Video-based	100.0%	0.0%	0.0%	100.0%	0.0%

Table 4.6: Classification results with simple affine face alignment based on AAM feature localizations

Furthermore, it is interesting to notice that the piecewise affine-based registration outperformed the affine-based approach with 1.0% of performance increase in terms of CCR. This indicates that the pose correction gained more confidence than the negative impact of non-linear piecewise affine warp on those frames with expression variations in the video sequences. The video-based classification yielded even better results, because the frame-level false decisions, i.e. false classifications and rejections were suppressed and only the final accumulated score were used for classification.

Frame-based classification indicates the system's performance on single frames, progressive score-based classification can be used in systems without fixed decision points, whereas video-based classification can be used in scenarios with fixed decision points, i.e. the decision is made when n_i frames of person i are recorded.

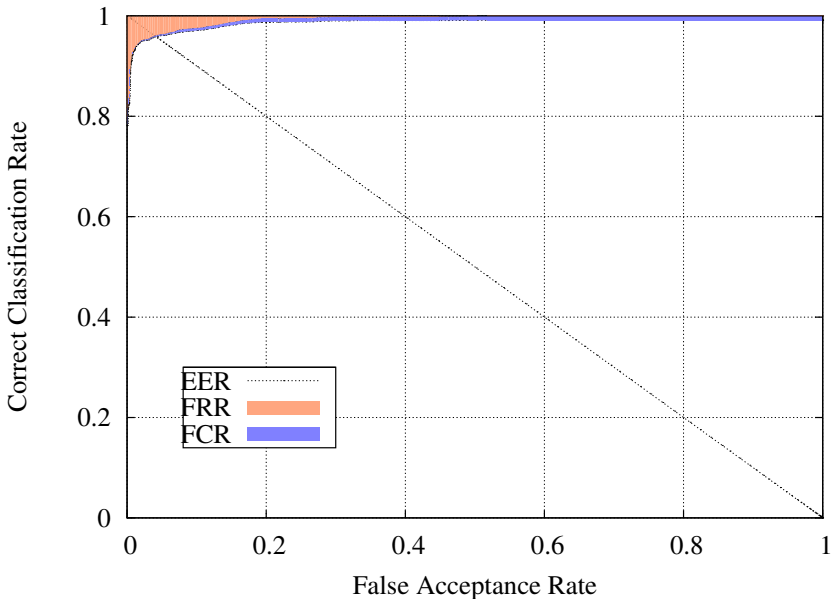


Figure 4.14: Frame-based receiver operating characteristics curve (AAM-based face synthesis)

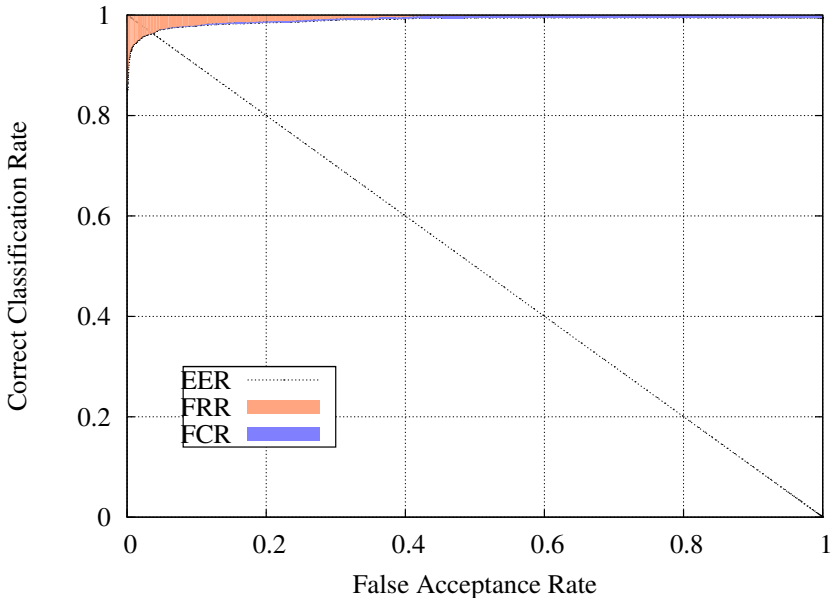


Figure 4.15: Frame-based receiver operating characteristics curve (Simple affine face alignment based on AAM fitting)

5 Conclusion

Face recognition techniques have been intensively investigated for decades aiming at high recognition accuracy and robustness against numerous facial appearance variations. Face registration is known as an important factor for recognition as demonstrated in many previous studies as well as in this study. The face registration is affected by factors such as illumination and pose variations. Moreover, due to the limitations of some simple registration approaches, the pose problem can not be solved at all. Another problem in many simple face registration approaches is that the quality of registration is not assessed, so that poorly registered face images are trained or tested which degrades the performance of the system.

In this thesis, a face registration approach based on AAM fitting was studied. The generated generic AAM modeled shape variations of face rotation in pitch and yaw angle so that pose information was obtained after model fitting. The pose of an input face was normalized with piecewise affine warp and a frontal view of the input face was synthesized. The modified histogram fitting approach was employed to mitigate the poor illumination problem. To initialize the model more precisely and robust against cluttered background, an inner-face AAM was built and the fitting was proceeded progressively. Face recognition was based on the fitted and pose normalized face images using local appearance-based approach. We extracted the local features from each block on a registered image using discrete cosine transform, and then concatenated the local features in order to conserve spatial information.

To evaluate the proposed face registration approach, we conducted three experiments on different databases. The first experiment was designed to evaluate the pose correction based on AAM fitting. We chose the $b*$ subset of FERET which contains various pose angles as well as illumination and expression variations in each of the series. Face recognition using AAM-based face registration achieved 97.8% of recognition rate on the data set with $\pm 15\,^{\circ}$ of face rotation in yaw angel. The simple affine warp approach however, only recognized 70.5% of the testing images with this pose angle. When the rotation angle increase to $\pm 60\,^{\circ}$, recognition based on simple affine warp could hardly recognize any images in the probe set, while the AAM-based approach still achieved a recognition rate of 48.3%. The proposed pose correction helped a lot to recognize faces with small angle rotation as we observed in this experiment. We also compared the local appearance-based face recognition to other holistic approaches such

as Eigenfaces and EHMM. The local appearance-based approach significantly outperformed the holistic approaches and it was more robust against the error introduced by AAM fitting and face synthesis.

In the second experiment, the accuracy of the facial feature localization was evaluated. We evaluated only eye localization due to the lack of ground-truth data for other feature points. The results were around average compared to other feature-localization approaches, since the AAM fitting is a holistic model matching approach, the precision of localizing a single feature point may be affected by the appearance of another part of the face. Reflection on glasses, for an example, might cause the localization failures.

In addition to evaluation on still images, we also carried out experiments on video sequences. Tracking and fitting AAM in video sequences are considered to be easier than fitting AAM on still images. Fitting a current frame can start from the model parameters fitted in the preceding frame. Fitted face appearance in preceding frame was utilized to compensate the mismatch between the input frame and the generic model. The modified DFFS metric was employed to assess the quality of fitting on a single frame. Open set face recognition was performed on the successfully registered frames, that is, the frames which were fitted with large DFFS values were discarded from the training and testing set. The results in this experiment showed that both pose correction and registration quality assessment improved the performance of open set face recognition compared to the previous system [61].

6 Future work

Further improvement can be made to make the proposed system even more robust and applicable.

In order to improve the fitting against partial occlusions, we plan to introduce the robust error function which is able to detect the partially occluded face region and suppress the impact of occlusion on the whole model fitting. As a byproduct, the detection of partial occlusion can also be used for weighting the local blocks for local appearance-based face recognition.

The single model AAM limits pose estimation especially in case of large angle rotation. As shown in the first experiment, the increasing yaw angle degenerates the fitting performance due to the local minimum problem and the mismatch between the warped face and the model appearance. To solve this problem, generating view-based AAMs is a promising solution as suggested in [17]. Full- and semi-profile faces are modeled separately from the near frontal faces which yields five AAMs that model different views of face. The face synthesis method should also be improved to avoid the over-sampling and down-sampling effect on semi-profile faces.

Another flaw of the current model is the compactness and size of the mean shape. The model is not compact enough and the mean shape size is too large so that each model fitting costs around $100 - 200$ ms. To make the face registration applicable in real time, a more compact model will be trained in the future, and the size of the mean shape will be shrunk so that less pixels are modeled for model appearance. However, we should also consider the trade off between the model resolution and the shape size.

Since the modified DFFS metric assesses the quality of model fitting, the DFFS values can be used to weight the frames contribution to the classification in a video-based face recognition scheme.

This work can be easily extended to head pose estimation, gaze estimation, facial expression analysis and gender recognition.

Bibliography

[1] "The BioID database," http://www.bioid.com/downloads/facedb.

[2] "The IMM face database," http://www2.imm.dtu.dk/˜aam/.

[3] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[4] D. Beymer and T. Poggio, "Face recognition from one example view," in *ICCV*, Boston, MA, 1995.

[5] A. Blake and M. Isard, *Active Contours*. Springer, 1998.

[6] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Siggraph 1999, Computer Graphics Proceedings*, Los Angeles, 1999, pp. 187–194.

[7] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. on PAMI*, vol. 25, no. 9, pp. 1063–1074, Sept. 2003.

[8] F. L. Bookstein, "Principal warps: thin-plate splines and the decomposition of deformations," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 6, pp. 567–585, 1989.

[9] D. Chai and K. N. Ngan, "Face segmentation using skin color map in videophone applications," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 9, no. 4, pp. 551–564, 1999.

[10] X. Chai, S. Shan, and W. Gao, "Pose normalization for robust face recognition based on statistical affine transformation," in *ICICS-PCM 2003*, vol. 3, Singapore, 2003, pp. 1413–1417.

[11] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[12] K. Chang, K. Bowyer, and P. Flynn, "Face recognition using 2D and 3D facial data," in *Proc. ACM Workshop on Multimodal User Authentication*, Dec. 2003, pp. 25–32.

[13] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," in *Proc. IEEE*, 1995, pp. 705–740.

[14] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," 5^{th} *European Conference on Computer Vision*, vol. 2, pp. 484–498, 1998.

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

[16] T. F. Cootes and C. J. Taylor, "Combining elastic and statistical models of appearance variation," in *Proc. of European Conf. on Computer Vision*, vol. 1, 2000, pp. 149–163.

[17] T. F. Cootes, K. Walker, and C. J. Taylor, "View-based active appearance models," in 4^{th} *Intl. Conf. on Automatic Face and Gesture Recognition*, Grenoble, France, 2000, pp. 227–232.

[18] P. Corcoran, M. C. Ionita, and I. Bacivarov, "Next generation face tracking technology using AAM techniques," *ISSCS 2007*, vol. 1, pp. 1–4, 2007.

[19] D. Cristinacce, T. Coote, and I. Scott, "A multi-stage approach to facial feature detection," in *Proc. British Machine Vision Conference*, vol. 1, 1996, pp. 277–286.

[20] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2001.

[21] G. J. Edwards, T. F. Cootes, and C. J. Taylor, "Face recognition using active appearance models," in *Proc. of* 5^{th} *European Conference on Computer Vision*, vol. LNCS-Series 1406-1607, 1998, pp. 581–595.

[22] H. K. Ekenel and R. Stiefelhagen, "A generic face representation approach for local appearance based face verification," in *CVPR Workshop on FRGC Experiments*, 2005.

[23] H. K. Ekenel and R. Stiefelhagen, "Local appearance based face recognition using discrete cosine transform," in 13^{th} *European Signal Processing Conference (EUSIPCO 2005)*, Antalya, Turkey, 2005.

[24] H. K. Ekenel and R. Stiefelhagen, "Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization," in *CVPR Biometrics Workshop*, 2006.

[25] H. K. Ekenel and R. Stiefelhagen, "Block selection in the local appearance-based face recognition scheme," in *CVPR Biometrics Workshop*, New York, USA, 2006.

[26] I. Fasel, B. Fortenberry, and J. Movellan, "A generative framework for real time object detection and classification," *Computer Vision and Image Understanding*, vol. 98, pp. 182–210, 2005.

[27] M. Fischer, "Automatic face retrieval in TV-series," Diplomarbeit, Interactive Systems Labs, Universität Karlsruhe (TH), June 2008.

[28] R. Fisker, "Making deformable template models operational," Ph.D. dissertation, Department of Mathematical Modeling, Technical University of Denmark, 2000.

[29] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *ISSCS 2007*, vol. 55, no. 1, pp. 119–139, 1997.

[30] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2001.

[31] C. Goodall, "Procrustes methods in the statistical analysis of shape," *Journal of the Royal Statistical Society, Series B*, vol. 53, no. 2, pp. 285–339, 1991.

[32] R. Gottumukkal and V. K. Asari, "An improved face recognition technique based on modular PCA approaches," *Pattern Recognition Letters*, vol. 25, no. 4, pp. 429–436, 2004.

[33] H. Greenspan, J. Goldberger, and I. Eshet, "Mixture model for face color modeling and segmentation," *Pattern Recognition Letters*, vol. 22, pp. 1525–1536, Sept. 2001.

[34] R. Gross, I. Matthews, and S. Baker, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 11, pp. 1080–1093, 2005.

[35] J. Guillemaut, J. Kittler, M. T. Sadeghi, and W. J. Christmas, "General pose face recognition using frontal face model," in 11^{th} *Iberoamerican Congress in Pattern Recognition*, vol. 4225/2006, 2006, pp. 79–88.

[36] S. Gutta, V. Philomin, and M. Trajkovic, "An investigation into the use of partial-faces for face recognition," in *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, 2002.

[37] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: Global versus component-based approach," in *ICCV*, 2001, pp. 688–694.

[38] M. Isard and A. Blake, "CONDENSATION – conditional density propagation for visual tracking," *Int. J. Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[39] T. S. Jebara, "3D pose estimation and normalization for face recognition," Bachelor's thesis, McGill Center for Intelligent Machines, McGill University, 1996.

[40] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," *Int'l J. Computer Vision*, vol. 46, no. 1, pp. 81–96, 2002.

[41] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. Journal of Computer Vision*, vol. 8, no. 2, pp. 321–331, 1988.

[42] D. Kim, J. Kim, S. Cho, Y. Jang, S.-T. Chung, and B.-G. Kim, "Progressive AAM based robust face alignment," in *Proc. of World Academy of Science, Engineering and Technology*, vol. 21, 2007, pp. 488–492.

[43] X. Liu, P. Tu, and F. W. Wheeler, "Face model fitting on low resolution images," in *Proc.* 17^{th} *British Machine Vision Conference*, vol. 3, Eiginburgh, UK, 2006, pp. 1079–1088.

[44] X. Liu, F. W. Wheeler, and P. H. Tu, "Improved face model fitting on video sequences," in *Proc. of British Machine Vision Conference (BMVC) 2007*, Warwick, UK, 2007.

[45] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of Image Understanding Workshop*, 1981, pp. 121–130.

[46] A. Martínez, "Recognizing imprecisely localized, partially occluded and expression variant faces from a single sample per class," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748–763, 2002.

[47] I. Matthews and S. Baker, "Active appearance models revisited," *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

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

[49] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object detection," in 5^{th} *Int'l Conf. on Computer Vision*, Cambridge, MA, 1995, pp. 786–793.

[50] A. Nefian, "A hidden Markov model-based approach for face detection and recognition," Ph.D. dissertation, Georgia Institute of Technology, 1999.

[51] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in 6^{th} *Int'l. Conf. on Computer Vision*, 1998, pp. 555–562.

[52] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, 1994.

[53] P. J. Phillips, "Support vector machines applied to face recognition," in *Advances in Neural Information Processing Systems 11*. MIT Press, 1998, pp. 803–809.

[54] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[55] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," *Image and Vision Computing*, vol. 16, no. 5, pp. 295–306, 1998.

[56] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 22–38, 1998.

[57] S. Sclaroff and J. Isidoro, "Active blobs: region-based, deformable appearance models," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 197–225, 2003.

[58] H.-Y. Shum and R. Szeliski, "Construction of panoramic image mosaics with global and local alignment," *International Journal of Computer Vision*, vol. 16, no. 1, pp. 63–84, 2000.

[59] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination and expression (PIE) database," in *Proc. IEEE Int'l Conf. Automatic Face and Gesture Recognition*, May 2002.

[60] M. B. Stegmann, "Active appearance models: Theory, extensions and cases," Master's thesis, IMM, Technical University of Denmark, 2000.

[61] L. Szasz-Toth, "Open-set face recognition," Studienarbeit, Interactive Systems Labs, Universität Karlsruhe (TH), Oct. 2007.

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

[63] V. N. Vapnik, *Statistical learning theory*. New York: John Wiley & Sons, 1998.

[64] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[65] G. Yang and T. S. Huang, "Human face detection in complex background," *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.

[66] J. Yang and A. Waibel, "A real-time face tracker," in *Proc. of Third Workshop Application of Computer Vision*, 1996, pp. 142–147.

[67] M. Yang, J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. on PAMI*, vol. 24, no. 1, pp. 34–58, 2002.

[68] A. Yuille, P. Hallinan, and D. Cohen, "Feature extraction from faces using deformable templates," *Int'l J.Computer Version*, vol. 8, no. 2, pp. 99–111, 1992.

[69] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys (CSUR)*, vol. 35, pp. 399–458, 2003.