Deformable 3D Face Tracking in Real World Scenarios

ZHANG, Wei

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
July 2010

Abstract

Three dimensional face tracking is a crucial task for many applications in computer vision. Problem like face recognition, facial expression analysis and animation, are more likely to be solved by if the geometry and appearance properties are available through a 3D face tracker.

In the first part of the thesis, the problem of tracking a non-rigid 3D face is studied. A novel framework for non-rigid 3D face tracking is proposed for applications in live scenarios. In order to extract more information of feature correspondences, the proposed framework integrates three types of features which discriminate face deformation across different views. The integration of these complementary features is important for robust estimation of the 3D parameters. In order to estimate the high dimensional 3D deformation parameters, we develop a hierarchical parameter estimation algorithm to robustly estimate both rigid and non-rigid 3D parameters. We show the importance of both features fusion and hierarchical parameter estimation for reliable tracking 3D face deformation. Experiments demonstrate the robustness and accuracy of the proposed algorithm especially in the cases of agile head motion, drastic illumination change, and large pose change up to profile view.

The video based face recognition is studied in the second part of the thesis. Compared to the still image based recognition methods, the video based methodsshare the merits of spatial temporal coherence among image sequences and overcomplete training samples. We propose a framework for the task of face recognition in real-world noisy videos based on 3D deformable face tracking, which can directly estimate face pose for a view-based face recognition scheme. Meanwhile, the precise non-rigid tracking provides well-aligned face samples for the subsequent recognizer. At the recognition stage, three types of feature descriptors, including Regularized LDA, LE and sparse representation, are exploited. Extensive experiments conducted on the real world videos demonstrate that the proposed recognition framework can achieve the state-of-the art recognition results, even with the usage of a simple classifier.

Finally, a performance driven face animation system is introduced. The proposed system consists of two key components: a robust non-rigid 3D tracking module and a MPEG4 compliant facial animation module. Firstly, the facial motion is tracked from source videos which contain both the rigid 3D head motion (6 DOF) and the non-rigid expression variation. Afterward, the tracked facial motion is parameterized via estimating a set of MPEG4 facial animation parameters(FAP) and applied to drive the animation of the target avatar.

摘要

在計算機視覺科學中，基於三維模型的人臉跟踪是一個關鍵的研究課題。通過人臉跟踪算法得到的人臉的幾何和表觀特徵，對解決人臉識別，表情分析和動畫生成等領域的很多課題，都有著重要的意義。

在論文的第一部分，我們著重討論了基於非剛體假設的人臉跟踪算法。對於現實場景中的人臉跟踪問題，我們提出了一個全新的算法框架。該框架包含以下幾個方面的要素：我們採用了三種不同的局部特徵抽取算法進行人臉特徵抽取，從而保證了在不同的視角下都能夠盡可能多的抽取用於匹配的二維特徵點集；同時，我們利用可形變的三維人臉模型來描述人臉的非剛性運動。在建立二維特徵點和三維人臉模型對應的基礎上，我們利用魯棒參數估計的方法來估計人臉的剛性運動參數和非剛性運動參數。在實驗部分，我們在大量的真實場景下的人臉運動視頻中測試了我們所提出的跟踪算法的有效性。對於真實場景中常見的極端狀況：包括快速的姿態變換，誇張的表情和不同的光照情況，我們都可以進行有效的跟踪。

基於視頻的人臉識別算法，是我們論文的第二個研究課題。和通常的基於圖像的人臉識別算法相比，視頻中的人臉識別問題有以下兩個方面的優越性：視頻中的人臉信息存在時域和空域上的約束性；基於視頻的訓練樣本對於特定的識別對象提供了完備的描述。對於現實場景中的人臉視頻識別問題，我們利用非剛體的人臉跟踪方法，分析每一幀圖像中的人臉的姿態，並對人臉特徵點進行精確的定位。通過精準的人臉抽取和對姿態進行分類，我們解決了人臉識別中常見的姿態變化和配準問題。在人臉識別階段，我們三種不同的特徵抽取算法和簡單的分類模式進行單幀圖像的人臉識別。為了測試我們的識別算法在真實場景中的有效性，我們對於大量從互聯網上所獲取的視頻進行識別實驗。和在相同數據集上的其他識別算法相比，我們所提議的識別算法獲得了大幅度的識別率的提升。

在論文的最後一部分，我們研究了基於表情驅動的三維人臉動畫生成。我們所提出的動畫生成系統包含瞭如下兩方面的要素：魯棒的三維人臉追踪和基於MPEG-4的表情生成系統。我們通過人臉跟踪算法，對驅動源中的三維人臉運動信息進行的有效抽取。在表情驅動階段，我們利用MPEG-4中表情參數的定義對三維人臉運動進行參數化求解，並利用遈些表情參數進行三維動畫的生成。

Acknowledgments

My first thanks go to my supervisor, Prof. Tang Xiaoou. I'm greatly impressed by his passion for research and patients for students. Xiouou has always been an enthusiastic advisor, providing encouragement, insight and a valuable big picture. His kind suggestions always guide me in the right direction. Prof. Liu Jianzhuang deserves the same thanks. He shows me the real meaning of hardworking. He's always conscientious and energetic, and helps me grow up in research. I'm so lucky to have such an excellent supervisor and such a nice professor.

I am and will ever be proud that I am a member of MMlab. We form a real family here. I am greatly indebted to Xiaodai, Huangting, Afeng, and Liu Ming. You're all great guys. I've been always enjoying the happy times here with you, doing sports, hiking, cooking, and shopping together. Moreover, you always stand by my side at any time whenever I need you. Thanks, my dear friends. You guys make my life here wonderful and colorful. Thanks Dahua for your help and kind suggestions. I also extend my thanks to the other MMlabers: Yueming, Pengfei, Yingze, Chunjing, Zhenguo, Yiwen, Deli, Chen Mo, Zhao Feng, Huanzi, Li Yun, Zhifeng, Xiaowei, Chen Yu, Kaiming, Zhimin, Boqing, Duhao, Liu Ke, Tianfan... We are a big family here and I'll always remember the time with you.

Last but not least, I thank my family, Mom and Dad for giving my life in the first place, ,for educating me, for your unconditional supports.

I love you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

One of the great challenges in computer vision is to build a system that can precisely track the facial motion. This task is difficult because the unpredictable nature of the facial motion. A face can appear in a variety of poses and expressions, and is often surrounded by clutter. Currently, most existing systems typically require strict assumptions, for example, near-frontal motion, unique illumination, or simplified background.

In the first part of the thesis, we focus on the problem of three-dimensional face tracking, a specific subfield of the research topic on face tracking, for the following potential advantages. Problems like face recognition, facial expression analysis, lip reading are more likely to be solved if a stabilized image is generated through a 3D head tracker. Determining the 3D head position and orientation is also fundamental in the development of vision-driven user interfaces, and, more generally, for head gesture recognition. Furthermore, head tracking can lead to the development of very low bit-rate model-based video coders for video telephone. Most potential applications for head tracking require robustness to significant head motion, change in orientation, or scale. Moreover, they must work near video frame rates. Such requirements make the problem even more challenging.

Many approaches have been proposed to recover 3D head motion. The first type of approaches are to use distinct image features [60][61][34][45],which work well when the features may be reliably tracked over the image sequence. When good feature correspondences are not available, tracking the entire head region using a 3D head model is more reliable. Both generic and user-specific 3D geometric models have been used for head motion recovery [22][29]. With precise initialization, such models perform well and introduce minimal error. However, when the initialization step is not good, model error will increase substantially and degrade motion recovery. To alleviate initialization errors, it is often an effective and robust way to use a much simpler geometric head model. Various planar model-based methods have been presented [6][3]. They model the face as a simple plane and use a single face texture to recover head motion. The approximation of a planar face model introduces trivial model error, which is less sensitive to small initialization errors. When the head orientation is not far from the frontal view, the planar assumption works well. To represent the geometry of the entire head, a more complete 3D model is necessary. In [10][5], an ellipsoidal model was used with good results for 3D head tracking. Cascia et al. [50] brought out a fast 3D head tracker that treats a 3D head as a texture-mapped cylinder. The head image is treated as a linear combination of a set of bases that are generated by changing the pose of a single face image. The head pose of the input image then is estimated by computing coefficients of the linear combination. Though simple and effective, the usage of of a single, static template is unable to accommodate the scenarios in which large out-of-plane rotation drives the face away from the camera. In [90][11][85], a more sophisticated 3D face was used to achieve high precision 3D tracking result, with the aid of robust statistical approaches for outlier rejection.

The previous mentioned approaches handle only rigid motion, where the non-rigid facial deformation is treated as errors via the robust techniques. Meanwhile, some specific vision tasks, for example, expression recognition, performance driven animation, have the requirement to analyze the detailed facial motion. Most of the non-rigid tracking algorithms have been applied for the 2D cases, since it is relative easy to acquire the descriptive representation like silhouette or contour. For the 3D case, a parametric model is often exploited to describe the non-rigid facial deformation. One of the representative work comes from [8], a linear deformation 3D shape model is used for the registration of faces. In [100], a similar linear 3D model is used to track the facial deformation caused by expression variation. However, these non-rigid tracking systems suffer from the generalization problem since the appearance feature they depend on is locally adaptive to the specific tracking target. They need special training for the tracking target to achieve a good result.

In Chapter 3, we propose a feature based framework for 3D tracking of the deformable face. In order to extract more discriminative information on face deformation across different views, we combine three types of features: the semantic features which are the same as those defined in ASM, the silhouette features which are dynamically specified and matched according to different poses, and the online tracking features which are obtained by the robust matching of image interest points. In addition to the feature augmentation, we show that hierarchical parameter estimation is quite important to progressively reject outlier feature correspondences and estimate both rigid and non-rigid 3D parameters. In our framework, the parameter estimation is formulated as an energy minimization problem under the constraint of a deformable 3D face model which usually converges in a few iterations, thus fulfills the real-time tracking task. With comparison to other non-rigid face tracking algorithm, the proposed tracking framework may have potential advantages in the following aspects. Firstly, it has a strong generalization power with the usage of local feature descriptors for face tracking. No special training is required for a specific tracking target. Meanwhile, no special parameter tuning work is required, all the tracking parameters are fixed for all persons.

In the second part of the thesis, we put our focus on the topic of video based face recognition, which is generally built up on a robust tracking subsystem. During the past decades, research topics on object recognition especially the biometric related recognition such as face are extensive developed. Despite recent efforts [114],[113],[74], accurate and robust face recognition remains a challenging task in live scenarios, where the major difficulties lie in the appearance variations arising from changes in pose, expression, illumination ,partial occlusion and motion blur. At the same time, video based face recognition has received significant attention in the community of computer vision since it can take full advantage of temporal coherence among consecutive frames to achieve more accurate recognition rate, where the weak recognition decision in each frame is integrated over the whole sequence.

For the tasks of video based face recognition, the first and principle problem to be addressed is to efficiently and effectively track the face for samples collection. It plays a key role for the success of recognition procedure. All the tracking for recognition approaches can be divided into categories by the tracking modules they used.

Due to the maturity of 2D face tracking techniques over the past decades, the majority of existing video face recognition approaches are based on 2D tracking [2],[48]. As for the state-of-the-art 2D face tracking approaches[48][77], appearance changes caused by face pose or viewpoint are learned by SVM [13], LDA [64], GMM [24] or a combination of such techniques. However, the tracking precision of these methods are limited with the rectangle described results, which contain a clutter background and cannot guarantee to provide well aligned faces. Although some techniques can be developed to reject bad 2D tracking results based on training a classifier on well cropped faces [48], they need a labor heavy work at the training stage where large amounts of samples are needed. Furthermore, view classification and appearance learning is significantly depended on the training data set and do not have good generalization capability for real-world applications.

3D tracking based methods belong to another category for video recognition, which try to recover rigid facial pose for the recognition task. In [31], person-specific textured 3D face model is used for face pose estimation across video frames. Promising results have been achieved with the usage of a simple confidence measure for characters recognition in featured movies. Similar 3D face model is used in [104] where 3D tracking is performed with illumination compensation. The main problems of above approaches are that considerable amount of user supervision is required to fit a textured 3D face model for tracking. Moreover, the using of holistic appearance template or model for face tracking does not work robustly in practice.

In Chapter 4, we propose a new tracking for recognition method which targets for applications in live scenarios. At the face tracking step, the related tracking module takes the advantage of 3D morphable model (3DMM) for non-rigid shape representation and uses local features set for facial appearance description, which include both off-line trained facial features and online tracked image features. Meanwhile, a hierarchical shape and pose estimation strategy is used for robust estimation of both rigid and non-rigid motion parameters. The integration of all the above ingredients ensures the tracking robust to occlusion, deformation and illumination changes, which are often encountered in the real world application.

Given the robust 3D tracking results, a simple recognition strategy is used at the recognition stage which consists of the following two aspects. Firstly, for each tracked frame, we use the widely used feature descriptors, for example, Regularized LDA [57], LE descriptor [14], sparse representation [97], for features extraction. Nearest Neighbor is used to selected the candidate identities; Secondly, for the whole tracked sequence, a simple frame fusion fusion scheme (voting scheme and distance fusion) is utilized to select a best matched identity. The confidence of 3D tracking is measured through the geometric consistency of facial features matching with the 3D model, which are used to select reliable frames for recognition. Given the extensive experiments on the public video dataset, we show that the proposed non-rigid tracking based recognition approach achieves significantly performance improvement over existing approaches, even with the usage of a simply holistic feature descriptor.

In the last part of the thesis, the problem of performance driven realistic facial animation is studied, which is still built up on the basis of non-rigid face tracking. The human face animation is a challenging task for both fields of research. In psychology, it has the demand for realistic, but controllable face stimuli. On the other hand, a good understanding of the cognitive processes of face perception in humans would clearly help Computer Graphics researchers and artists in the difficult task of synthesizing realistic virtual humans. With the successful of vivid facial animation generation, a variety of industrial applications could benefited: computer games, human-computer interfaces (HCI), teleconferencing, medical rehabilitation systems, computer based training and consulting as well as the film industry.

In order to achieve the goal of realism facial animation acquisition, the real-world facial data is used extensively. Shape and appearance information of a face and its deformation are measured in a 3D scanner and then converted into a morphable 3D face model. Additionally, motion information for a sparse set of facial markers is acquired using an optical Motion Capture system. The deformation of the 3D model are computed from motion capture data by decomposing the marker trajectories into semantically meaningful motion elements based on the Facial Action Coding System (FACS) [28], which defines a set of basic facial motions called Action Units (AUs). These AUs approximately correspond to natural muscle activations, providing an intuitive and accurate system for annotating facial motion. Using FACS as a basis has the following two additional advantages. Its semantics allow easy retargeting of the motion onto any face model that uses the same semantic structure. In contrast to approaches that use statistical concepts such as Principle Component Analysis, AUs can be verbally described. Thus, matching facial expressions can be generated by actors or artists. Furthermore, AUs describe local effects in the face which is beneficial for a generative model of facial motion. It has been argued that AU activations might fail to describe the facial state accurately since they might reflect the combined activations of multiple muscles and do not take temporal information into account [30]. However, most of the above animation systems are built on the basis of a complicated motion capture subsystem, which make it infeasible to be applied on a low-end PC.

In Chapter. 5, we present a system for realistic facial animation that transfers the captured facial motion into semantically meaningful expressional channel based on the MPEG-4 standard. The proposed animation system consists of two key components: a robust non-rigid 3D tracking module and a MPEG4 compliant facial animation module. Firstly, the facial motion is tracked from source videos which contain both the rigid 3D head motion (6 DOF) and the non-rigid expression variation. Afterward, the tracked facial motion is parameterized via estimating a set of MPEG4 facial animation parameters(FAP). As the final step, these FAP values are transferred to the MPEG4-compliant face model for the animation purpose. The proposed tracking and animation system has a strong generalization ability and can be used in the indoor environment with no additional assumptions.

1.1 Thesis Framework

The rest of thesis is organized as follows. In Chapter. 2, we give a brief view of the tracking related work. The proposed non-rigid tracking framework is presented in Chapter. 3. In Chapter. 4, the video based face recognition is introduced with the experiments results. Chapter. 5 shows the system of performance driven face animation. In Chapter. 4.5, we conclude the thesis and highlight the future work.

Chapter 2

Related Work

There is an abundance of visual tracking work in the literature, from a simple template matching approach [38] to a 3D model-based algorithm [100]. These algorithms differ mainly in the representation scheme: ranging from color pixels, blobs, texture, features, image patches, templates, active contours, snakes, wavelets, eigenspace, to 3D geometric models; and in the prediction approach, such as correlation, sum of square distance, particle filter, Kalman filter, EM algorithm, Bayesian inference, statistical models, mixture models, and optimization formulations. A thorough discussion of this topic is beyond the scope of this thesis. In this chapter we review only the most relevant object tracking work and focus on the algorithms that operate directly on gray scale images.

2.1 Template Based Tracking

Hager and Belhumeur [38] propose a tracking algorithm, using parametric motion models, which holds a constant reference template. To avoid computing the whole Jacobian matrix at every iteration, it is factored into a constant part that is depended on the brightness gradients of the reference template multiply the spatial derivative of the transformation, and a variable part consisting of the derivative of the transform with respect to the motion parameters. Illumination is handled by learning a subspace from training samples taken from various lighting conditions and using it to precompute a constant correction matrix. M-estimation, implemented using iteratively reweighted least-squares, is used to handle partial occlusion.

Shi And Tomasi present a local feature tracking algorithm for identifying and tracking a set of feature points [76]. Their system computes only the translational component of motion during tracking. In an attempt to solve the drift problem, the features are monitored during tracking by warping them back to the original frame using an affine motion model. The similarity between the feature points in the current and original frames is computed via the sum of squared differences (SSD). If the dissimilarity of a feature between the original and current frames becomes too huge, then that feature is discarded.

Tracking with fixed template can be reliable over short duration, but it is poorly to handle with appearance during a long tracking period, which is common in most scenarios. One can improve the robustness of such trackers by representing the variability of each pixel in the template. This allows the tracker work efficiently against a clutter background.

2.2 Subspace Based Tracking

Tracking robustness can be further enhanced with the use of subspace models of appearance. Such view-based methods, usually learned with Principle Component Analysis (PCA), have the advantage of modeling variations with lighting and pose. However, they still have the disadvantages that they are object specific and require that training prior to tracking for the appearance subspace learning.

In [7], two contributions have been made to the subspace-based tracking. First, matching between the eigenspace and the image is made robust by replacing the eigen-coefficients with a robust norm error functions that are robust to outliers arising from occlusions, background clutter, and noise. Second, they introduce the subspace constancy assumption, which is a generalization of the notion of brightness constancy used in optical flow.

Cootes *et al.*[21] introduced Active Shape Model (ASM) as a method for representing and searching for 2D shapes. An ASM is obtained from a training set which has been annotated with landmark points. The training set is first aligned by Procrust Analysis. Thereafter, a mean shape and a matrix containing the main modes of variation about the mean shape are determined using PCA. Any particular shape from the class can be expressed as a combination of the mean shape plus a set of shape parameters times the modes of variation. The shape parameters have a zero mean Gaussian distribution. New shape similar to those from the class can be generated by varying the shape parameters with a range of a few standard deviations. Searching a new image using an ASM is an iterative process, consisting of placing the ASM in the image and then moving the points towards the strongest edges under the constraint that shape parameters remain within the subspace and have a Gaussian prior.

Cootes *et al.*[20] later introduced Active Appearance Model (AAM), which extends the local feature model with a shape-free appearance model. The appearance model is obtained by warping the training image to the mean shape and then apply PCA to the extracted textures. The shape and appearance parameters are then concatenated into a single vector. Allowing for different in units and a further PCA is applied on this data to obtain a combined model that controls both shape and appearance. AAMs can be used to generate photo-realistic reconstructions of objects like faces, and varying the appearance parameters within a few standard deviations can generate novel images from the same class. As with ASMs, an AAM search consists of placing a model in an image and then changing the appearance parameters under the constraint that they remain within the subspace and have a Gaussian prior. Both ASMs and AAMs have been used for tracking. For instance, during tracking one would expect identity to remain constant while expression, pose and lighting vary with their own dynamics. The shortcoming of this approach is that these residual variations are class-specific.

2.3 3D Model Based Tracking

The common 3D model based methods assume that a 3D model of the target object is available a priori. The task of tracking is thus reduced to registering the model with the image of the target in the scene. These approaches can be divided into two categories by their tracking precision.

2.3.1 Rigid 3D Tracking

In the case of only rigid motion is considered, the most common strategy is to use simple geometric head models. Xiao *et al.* [101] propose a 3D face tracking algorithm, which recover the rigid motion of the head from an input video using a cylindrical head model. It recovers the global motion by minimizing the difference of texture or optical flow between observation and the model. As a template matching based approaches, the proposed system builds up a set of reference template online and register the input frame to the registered templates dynamically. To handle the non-rigid motion and occlusion,the iteratively re-weighted least squares (IRLS) technique is used for registration error norm minimization. To accommodate the self-occlusion and lighting variation, the templates are dynamically update using robust techniques. Similar head geometry and matching strategy is used in [50]. However, the illumination is compensated by the precomputed illumination basis vectors. Vacchetti *et al.* [86] used multiple key frames and feature point matching to estimate the motion of their model under large pose variation. These approaches assume that the 3D shape of the object does not change during tracking, they do not need to handle the non-rigid deformation.

For all the rigid 3D tracking, the number of all the model parameters are at most 6-dim. The low dimensionality of the parameter space results in robust tracking performance when compared to the high dimensionality of the AAMs. In addition, these methods do not require any learning stage, which means that they are person independent. Moreover, they are robust to a large pose change because they use the whole area of the head in the image instead of a specific part.

2.3.2 Non-rigid 3D Tracking

Some researchers tried to track the deforming shape and global motion at the same time. Structure from motion is a typical band of approaches which directly derive the 3D object structure from the input image stream. Bregler *et al.* [9] proposed a factorization method to simultaneously reconstruct the non-rigid shape and camera projection matrices. This method was extended to a trilinear optimization approach in [83]. The optimization process involves three types of unknowns, shape vectors, shape parameters, and projection matrices. However, these factorization based approach require a high quality sparse features correspondence during the whole tracking sequences. Otherwise, robust techniques are required to reject outliers for precious structure acquirement[12].

When the prior of tracking targets deformation is available, DeCarlo and Metaxas [22] used a deforming face model whose fitting algorithm integrated optical flow and edge information. $2D + 3D$ AAM [100] is flexible model that can explain the varying shape and appearance of a non-rigid object. It is especially useful when we are interested in describing a specific part of a 3D object and the part is always visible because the AAM require that the topology of the shape must be consistent. 3D morphable models (3DMMs) [8] is another type of flexible model that are very similar to the 2D+3D AAM that are extension of the traditional 2D AAMs by incorporating 3D shape models into themselves. They require a large number parameters to describe the detailed variations of the shape and appearance and lack of generalization since precomputation of appearance is required. Meanwhile, these algorithms are not adequate for tracking of large head pose change and the quality of the estimated 3D shapes cannot be guaranteed.

Chapter 3

Non-Rigid 3D Face Tracking

In this chapter, we develop a novel framework for 3D tracking of the non-rigid face deformation from a single camera. The difficulty of the problem lies in the fact that 3D deformation parameter estimation becomes unstable when there are few reliable facial features correspondences. Unfortunately, this often occurs in real tracking scenario when there is significant illumination change, motion blur or large pose variation. In order to extract more information of feature correspondences, the proposed framework integrates three types of features which discriminate face deformation across different views:

- the semantic features which provide constant correspondences between 3D model points and major facial features;
- the silhouette features which provide dynamic correspondences between 3D model points and facial silhouette under varying views;
- the online tracking features that provide redundant correspondences between 3D model points and salient image features.

The integration of these complementary features is important for robust estimation of the 3D parameters. In order to estimate the high dimensional 3D deformation parameters, we develop a hierarchical parameter estimation algorithm to robustly estimate both rigid and non-rigid 3D parameters. We show the importance of both features fusion and hierarchical parameter estimation for reliable tracking 3D face deformation. Experiments demonstrate the robustness and accuracy of the proposed algorithm especially in the cases of agile head motion, drastic illumination change, and large pose change up to profile view.

3.1 Motivation

Reliable tracking of 3D deformable faces is still a challenging task in computer vision. The basic difficulty is that we need to estimate dozens of rigid and non-rigid 3D parameters from noisy image observation. The situation becomes even worse when the face is in profile view and almost half of the face region is occluded.

There are mainly two categories of 3D deformable face tracking algorithms. The first one is appearance based approach [100] which uses generative linear models of face appearance such as 2D Active Appearance Models (AAM) [65] and 3D Morphable Models [8] to capture the texture and shape variation of face respectively. The deformation parameters can then be efficiently estimated using gradient descent optimization. However, it is known that AAM has quite weak generalization capability because the general face texture space is too large to be captured by reasonable size training data. Therefore such approaches are limited to be used in conditions similar to the training data. The second one is feature based approach which uses aggregation of sparse facial features to represent face. The face variation is captured either by a multiple view 2D Active Shape Model(ASM) [115] or a complete 3D shape model [36] or a combination of both [89]. Compared with the appearance based algorithms, the feature based algorithms have better generalization capability. However, the local features used in ASM are semantic features whose correspondences in 3D model are manually defined. They are mainly located in regions around eyes, nose, and mouth which could be severely occluded when the face is in poses with pan angle larger than 45 degrees. However, the shortage of reliable semantic correspondences could make the tracking unstable.

In this chapter, we propose a feature based framework for 3D tracking of the deformable face. In order to extract more discriminative information on face deformation across different views, we combine three types of features: the semantic features which are the same as those defined in ASM, the silhouette features which are dynamically specified and matched according to different poses, and the online tracking features which are obtained by the robust matching of image interest points. In addition to the feature augmentation, we show that hierarchical parameter estimation is quite important to progressively reject outlier feature correspondences and estimate both rigid and non-rigid 3D parameters. In our framework, the parameter estimation is formulated as an energy minimization problem under the constraint of a deformable 3D face model which usually converges in a few iterations, thus fulfills the real-time tracking task.

3.2 Algorithm Overview

Figure. 3.1 shows the flow diagram of the tracking algorithm at frame t: given the input image, the algorithm builds up the feature matches and estimates the 3D pose and deformation parameters in a hierarchical way: Firstly the initial 3D pose parameters are estimated from online interest point information, then the initial feature matches are refined by rejecting many outliers and the shape and pose parameters are estimated in an iterative way. Since the shape and 3D pose estimation are based on feature matching between the neighboring frames, it is well known that such a strategy always suffers from accumulated drift. Our solution is the usage of key-frames and to anchor the current frame to them. The key-frames are selected online from the past estimated frames

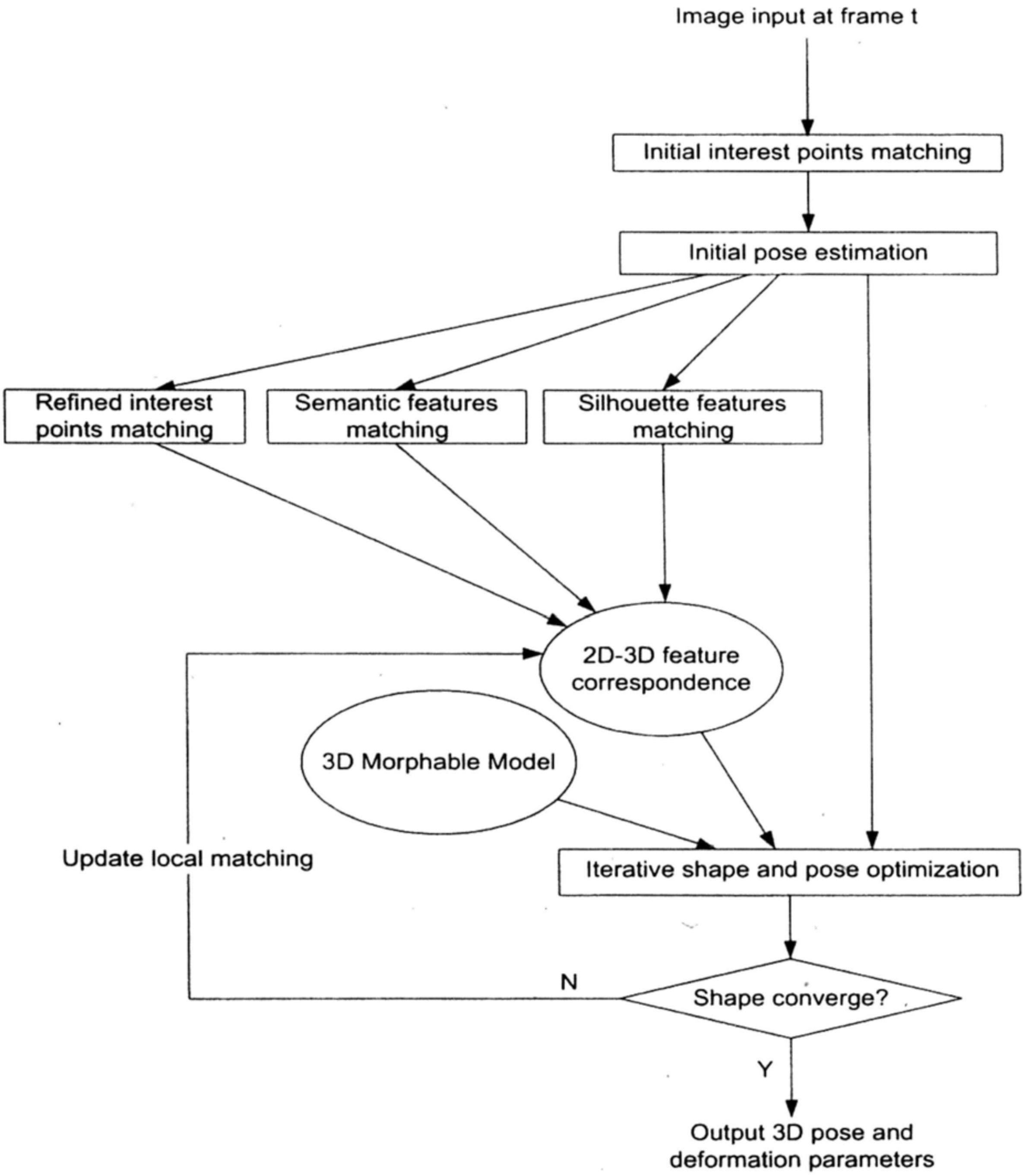


Image input at frame t
Initial interest points matching
Initial pose estimation
Refined interest points matching
Semantic features matching
Silhouette features matching
2D-3D feature correspondence
3D Morphable Model
Update local matching
Iterative shape and pose optimization
N
Shape converge?
Y
Output 3D pose and deformation parameters

Figure 3.1: Flow diagram of the proposed algorithm

with high accuracy in which pose and shape parameters are deterministic. For the details of key-frame selection and matching, we take the similar strategy to [90], the nearest best matched frames are chosen to anchor the current frame.

Initial Pose Estimation: The initial 3D pose estimation is obtained from interest points matching among the current frame, previous frame and some selected key-frames, the 2D interest points in previous frame and key-frames can be back-projected into the 3D face model to get their 3D position. The initial pose in current frame can be estimated by the 2D-3D point correspondences in a Bayesian inference way. Details are given in Section. 3.8.1.

Iterative Shape and Pose Optimization: Given the initial pose estimation, the matches for semantic features, silhouette features, and interest points are obtained and refined, their 3D correspondences in the model are also determined. Then the 3D pose and deformation parameters are estimated from the 2D-3D point correspondences. The above procedure is carried out iteratively until convergence is achieved. Details are given in Section. 3.5.

Tracking Initialization: In order to make the whole tracking algorithm fully automatic, we need to estimate the 3D pose and deformation parameters in the first frame: the face is first detected by the face detector [102], then the facial features are extracted by the face alignment algorithm [105]. Such facial features are semantic features whose 3D correspondences in the face model are fixed, so we can apply the POSIT algorithm [23] to estimate the 3D pose using rigid semantic features and then estimate the 3D deformation parameters.

3.3 3D Deformable Shape

The shape of a face is defined by a 3D triangulated mesh. The shape vector is denoted as $S = (v_1^t, \ldots, v_n^t)$, where $v_i = (x_i, y_i, z_i)^t, (i = 1, \ldots, n)$ are the 3D

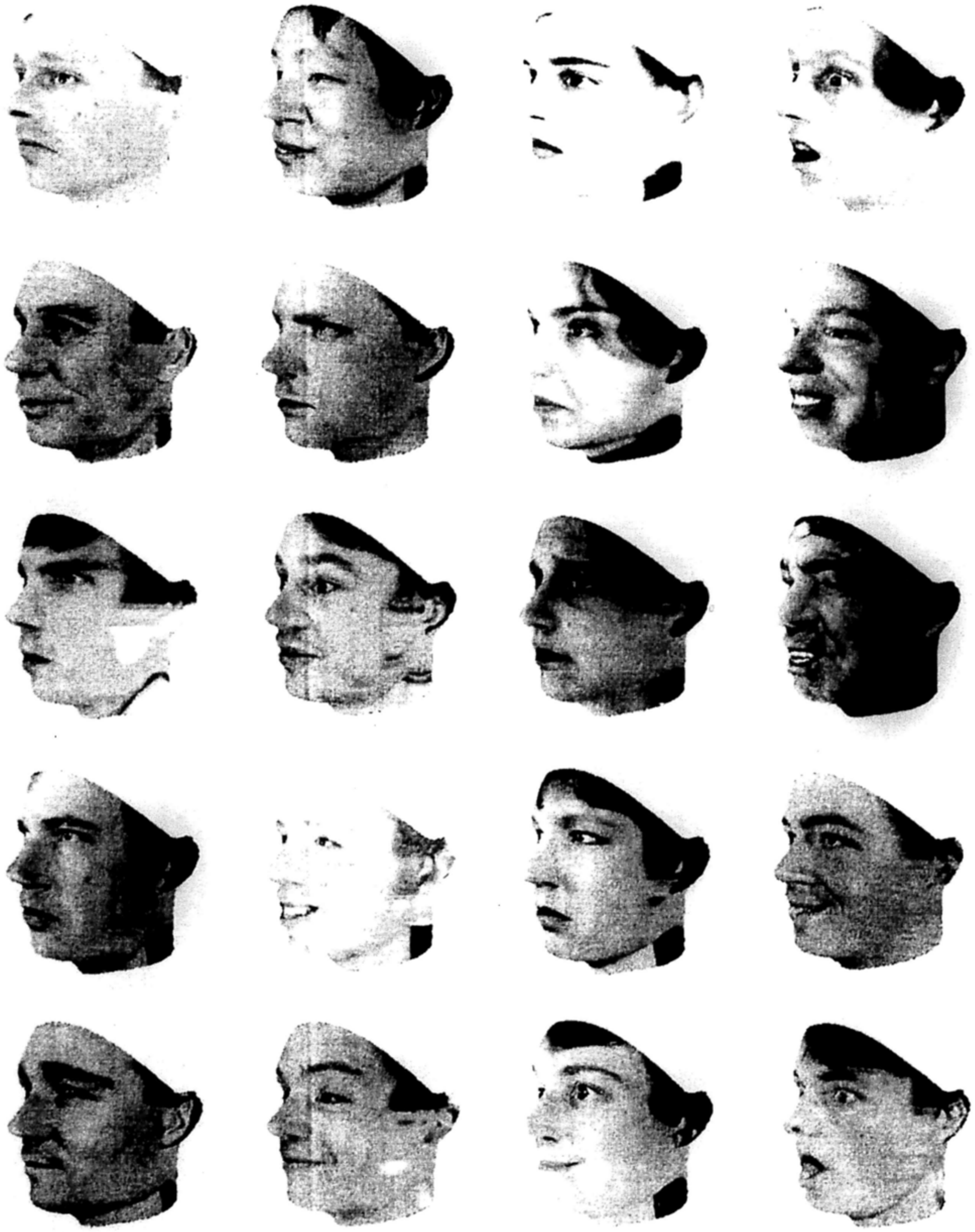


Figure 3.2: Synthesized 3D face model with expression

coordinates of the i-th vertex. The deformation of a 3-D face is described by a linear model [8][100][103], such as PCA:

$$S = \mu + \sum_{i=1}^{k} \alpha_i \Phi_i, \tag{3.1}$$

where μ represents the average shape from the training samples, Φ_i are orthogonal shape vectors, and α_i are scalar values indicating the contributions of the shape deformation from the i-th shape vector.

3.3.1 Training 3D model from image set

Collecting and labeling a large number of 3D faces is itself a difficult problem. Several techniques [8],[111] have been developed to establish the correspondence among 3D laser scans automatically. However, no guarantee can be made as to the correctness of the correspondence with the usage of optical flow. In our work we adopt a different strategy: use synthetic 3D faces instead of real faces for training the deformable shape model and local patterns.

Our experiments involve two face set: A) Mixture of face images from AR Database [63] and XM2VTSbd Database [66]: 2907 frontal images, each image is manually labeled with 83 landmarks. B) USF Human-ID Database [8]:100 laser scanners aligned to a 3D reference model of 8895 points. We create a synthetic 3D face database from A and B for training: Recall that the 2D face model uses 83 points, and each image in set A is labeled with those landmark points. In the meantime, each 3D laser scanned face in set B are marked with 8895 reference points. Therefore once we establish manually the correspondences between those 83 points and 8895 points, we know the texture mapping between any pair of image I in database A and 3D face L in database B. A new virtual 3D face is generated automatically from I and L with the process in Figure. 3.3.

Firstly, we compute the relative 3D pose of I with respect to L that minimizes their shape and post difference iteratively on the image plane; Secondly,

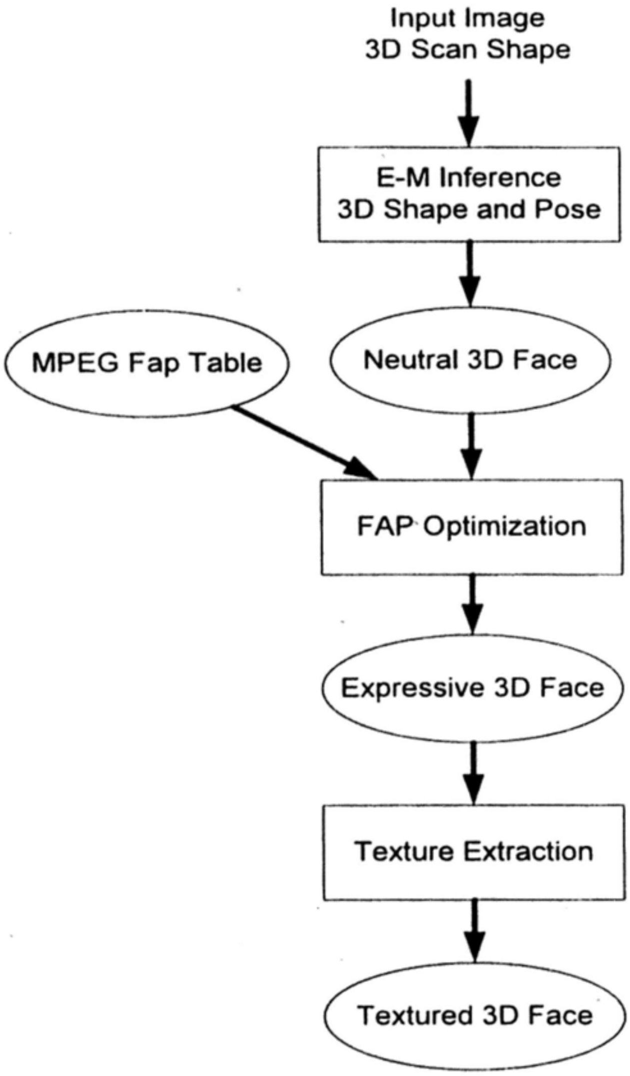


Figure 3.3: Diagram of 3D Face Synthesis

we utilize the MPEG Facial Animation Parameters (FAP) for the expression parameters extraction; At the last step, we extract texture from the 2D image. If holes (self occlusion) exist, the interpolation technique is utilized to fill the missing entry.

Repeating this process for every image in databases A and 3D shape databases B, we produce a gigantic database of over 2907 synthetic 3D faces with established correspondences. Observed from Figure. 3.2, the synthetic 3D faces may or may not correspond to a real person, but they are all plausible face instances. Even with texture holes exist, the reconstructed 3D face model can be used to train deformable shape model and local appearance models without loss of performance.

The original 3D face models have 8955 vertices and 17535 triangular facets which are too many for the 3D tracking task in a relatively low resolution video sequence, so we simplify the original models which contain only 180 vertices and 332 triangular facets. The 3D deformable face model is trained on the simplified models via PCA, where $42-$dim shape variations are kept with 99% energy preserved. In the experiment section, we will show that such a low dimensional representation will handle the expression variation among different persons effectively.

3.4 Local Features Matching

There are three types of local features used in the proposed algorithm for 2D-3D features correspondence seeking: the semantic features are defined in inner face region, such features have fixed correspondences in 3D model which does not change with 3D face pose; the silhouette features locate at the boundary of the face region and their 3D correspondences are dynamically determined according to the 3D face pose; the interest points are salient image features on the face, so they vary across images. The appearance of the first two type of features have a fixed pattern so that they can be learned from training samples in offline training stage. The third type of features are selected online by the image saliency. In the following subsections, we will describe the training and matching methods of all these local features.

3.4.1 Discriminative Learning for Type I,II Local Features

The local appearance model can be learned from a given set of example images with ground truth annotations. The task of learning is to learning a fitting function (model) that best fitting an image. In most of the existing work, generative models are commonly used in learning; examples of generative models include ASM [21], Gaussian Mixture Mode(GMM) [115]. Generative models learn the relationship between the ground truth feature point and their appearances to characterize the intrinsic generating process of feature pattern. However, a generative function does not typically represent the background and therefore not optimal to distinguish the ground truth feature points from their background: The local features associated with the neighborhood points of a feature point are often similar to that associated with the feature point; moreover, the most representative features are not always the best discriminative features.

Given sufficient training examples, discriminative learning approaches can provide better fitting functions. Discriminate learning has been applied successfully to object detection applications [88][59][54], in which the problem is formulated as a classification problem. In the training stage, the image patches containing the ground truth points are considered as positives, and the patches containing the points from neighborhood are treat as negatives. In testing, the location of the feature point is located by scanning, using the trained classification function, the test image over an exhaustive range.

Recently, discriminative learning has been incorporated into generative models to improve the pattern localization performance. In [105], the local pattern detectors are trained to replace the generative models in ASM for better locating the facial components. In [54], view based classifiers are plugged into the 3D post estimation to provide the evidence of whether the current feature belongs to the target object or not. The models built by these approaches improve the localization results. However, they could still have local extremes.

Discriminative learning via ranking is originally proposed to retrieve information based on user preference. It has been widely used in the vision tasks recently, like shape registration [98], deformable shape segmentation [106]. In this section, we utilize a supervised learning algorithm, RankBoost[33], to build the local feature model that ensures the pattern around the ground truth position will more likely have a higher confidence output than its neighbors.

The reasons that we prefer to rank learning for the local appearance model other than classification and regression are as following:

- Instead of learning a classification function that will discriminate all patterns around the ground truth from all patterns around their neighbors, our target is to learn a measurement(function) that only need to satisfy partial constrains on local pattern from one ground truth only with its neighbors, which meets the criterion for rank learning quite well.

- Unlike the regression based approach, which enforces a regressor to produce continuous output in the model space, ranking only tries to learn partial relations of paired points from the samples. The less constraints on the continuous in the model space makes ranking to be more generative.

A detailed comparative study on the relationship between these learning approaches can be referred to [106].

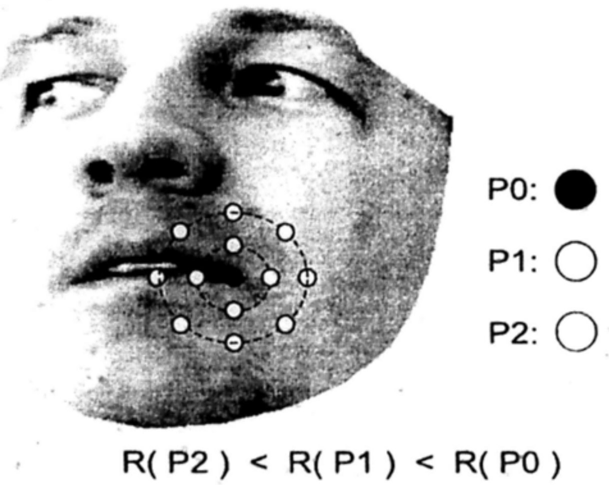


Figure 3.4: Rank Preference on the Facial Pattern

In the rank learning problem, we have a set of samples $\{\Phi_g, \Phi_n\}$, in which $\{\Phi_g\}$ denotes the samples collected from the ground truth position, $\{\Phi_n\}$ denotes the samples collected from neighbors. As shown in Figure. 3.4, let $\{P_0, P_1\}, \{P_1, P_2\}$ be pairs of point candidates and its associated feature pairs $\{x_0, x_1\}, \{x_1, x_2\}$. The ordering of $\{x_0, x_1\}, \{x_1, x_2\}$ is determined by their relative distance to the ground truth: a point P_0 that is the ground truth has a higher rank than a point P_1 from faraway.

Mathematically, the learning task for feature ranking is to learn a ranking function R that satisfy the following constraints:

$$\begin{cases} R(x^1) < R(x^2) & ||p^1 - \hat{p}|| > ||p^2 - \hat{p}|| \\ R(x^1) = R(x^2) & p^1 \ \textit{unrelated} \ p^2 \end{cases} \tag{3.2}$$

The RankBoost algorithm contains a set of *weak* ranking evaluation functions $F = \{f_i, i = 1, \ldots N\}$, where N is the number of *weak* ranking function. Based on different ranking tasks, the ranking features used in *weak* ranking function is different. In the proposed tracking system in this thesis, we define a new ranking feature f_i on the eigen-vector of the PCA model. Suppose we can train a PCA (generative) model $S = \{\mu, \Phi, \Lambda\}$ of the local patches when given a set of ground truth patches. For a new input local patch, we can evaluate

```
REQUIRE:  Initial distribution $\mathbf{D}$ over sample pairs $\boldsymbol{\Omega} : \boldsymbol{\varphi} \times \boldsymbol{\varphi}$
RETURN:   Final ranking function: $R(x) = \sum_{t} \alpha_t \mathbf{f}_t(x)$

INITIAL:  Set $\mathbf{D}_1 = \mathbf{D}$
    for $\mathbf{t = 1}$ to maximum round do
        Training all weak learners $\{\mathbf{g}_i\}$using the distribution $\mathbf{D}_t$ .
        Evaluate each weak learner$\mathbf{g}_i$ by the rank loss function:
            $C(\mathbf{D}_t, g_i) = \sum_{(x0,x1)\in\Omega} \mathbf{D}_t(x_0, x_1)(g_i(x_1) - g_i(x_0))$
        Select the weak learner$\mathbf{f}_t$ with maximum rank loss:
            $\mathbf{f}_t = \arg\max_{g_i} |C(\mathbf{D}_t, g_i)|, \varepsilon_t = |C(\mathbf{D}_t, \mathbf{f}_t)|,$
        Set $\alpha_t = \frac{1}{2}\left(\ln\frac{1+\varepsilon_t}{1-\varepsilon_t}\right).$
        Update the sample pairs weight:
            $\mathbf{D}_{t+1}(x_0, x_1) = \frac{\mathbf{D}_t(x_0, x_1)\exp(\mathbf{f}_t(x_0) - \mathbf{f}_t(x_1))}{Z_t},$
        $Z_t$ is a normalization factor that enforce $\mathbf{D}_{t+1}$ to be a distribution.
        Add the weak learner into the final ranking function:
            $R_t(x) = R_{t-1}(x) + \alpha_t \mathbf{f}_t(x)$
    end for
```

Figure 3.5: Train Local Feature Models with RankBoost

the likelihood $p(x|S)$ of the input patch x:

$$p(x|S) \propto \exp(||\Phi^T(x-\mu)||_\Lambda) = \prod_{i=1}^{N} f_i(x). \tag{3.3}$$

The geometrical interpretation of the *weak* ranking function f_i is the projection value of x to the $i-$th eigen-vector of the PCA model S.

The boosting algorithm uses a set of *weak* ranking functions F to update the distribution as shown in Figure. 3.5. Suppose that $\{x_0, x_1\}$ is a crucial pair so that we want x_1 to be ranked higher than x_0. Assuming for the moment that the parameter $\alpha_t > 0$, this rule decreases the weight $D_t(x_0, x_1)$ if f_t gives a correct ranking:$f_t(x_0) > f_t(x_1)$. Otherwise, the pair weight will be increased. Thus, $D_t(\cdot)$ will tend to concentrate on the pairs whose relative ranking is hardest to determine, which is the basic mechanism of the boosting family approaches.

Based on the definition Eq. (3.3), the *weak* ranking function f_i satisfies that:

$$0 \le f_i \le 1. \tag{3.4}$$

Based on the conclusion in [33], when we choose the weighted rules $\alpha_t = \frac{1}{2}\left(\ln \frac{1+\varepsilon_t}{1-\varepsilon_t}\right)$, the rank loss function Eq. (3.5) is bounded.

$$\begin{gathered}\sum_{(x_0,x_1)\in\Omega} D(x_0,x_1)\lceil R(x_1) \le R(x_0)\rceil \\ \lceil x\rceil = \begin{cases} 1, & x \ is \ true \\ 0, & x \ is \ false \end{cases}\end{gathered} \tag{3.5}$$

By taking into account the neighborhood and enforcing a rank prior in the learning stage, the discriminative local appearance model has a better localization ability with comparison to the traditional PCA model. We conducted experiments to compare the performance of the proposed models using RankBoost with that using the traditional PCA models. We use the ratio of of correct rank pairs to the total pairs as an index to evaluate the localization power of the proposed models.

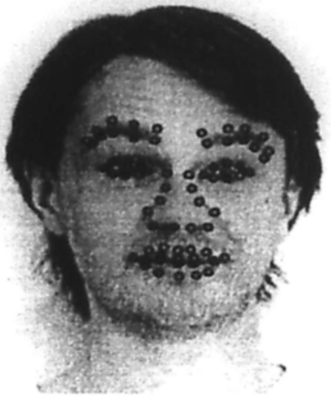


Figure 3.6: Annotated Image with 68 Feature Points

Given an image set of 721 images, where each image is annotated with 68 features points as shown in Figure. 3.6. For each annotated feature point, we collect a 9 × 1 image patch centered at ground truth. Meanwhile, we collect a 9 × 5 neighbor regions around the ground truth with a fixed stepsize of 2 pixels. These collected patches are constructed to rank pairs by their relative position to the ground truth. To take the comparison into a unified framework. For the PCA model, we take the Mahalanobis distance as the rank function.

Figure. 3.7 give the comparison result for all 68 features points. The proposed rank models outperform the traditional PCA models in all the points, while the maximum accurate rate over the traditional PCA models is more than 10 percentage.

3.4.2 Silhouette Feature Matching

For the semantic features and silhouette features, the process of their offline training stage are mentioned in Section. 3.4.1: they both have annotated ground truth in the training images. However, the silhouette features are slightly different from semantic features since their 2D positions and 3D correspondences are dynamically changed with respect to face pose. In the following

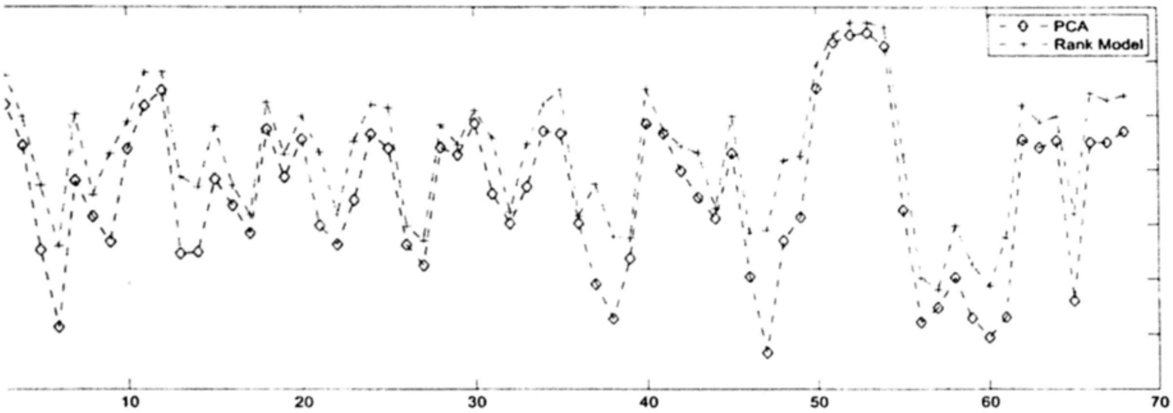


Figure 3.7: Localization Power Comparison: Rank Model V.S PCA

subsection, we describe our method to determine of the 2D-3D correspondences between silhouette features and 3D face model. The whole process for silhouette feature seeking and correspondence determination is performed by the following algorithm that include four steps with the results, on an example image, is shown in Figure. 3.8.

1. *Mesh map:* First a mesh map is constructed. A mesh map is a rendering of the face, but instead of setting the R, G, B color at one pixel, a index of a facet which covers the pixel is set. If more than one facet cover the same pixel, the index of facet which has the minimum depth value is kept. Such a process is similar to the Z-buffer algorithm in computer graphics.

2. *Binary map:* The mesh map is converted to a binary map, in which a value of 0 is for a pixel in the face area, and 1 is for a pixel outside the face area.

3. *Contour Map*: Since face region in a binary map has good connectivity (no holes). A scanline algorithm [99] is used to determine the contour map. Each pixel on this contour map set to 1 is on the contour of the face.

4. *Silhouette Features* Sampling the contour map with a fixed stepsize. For

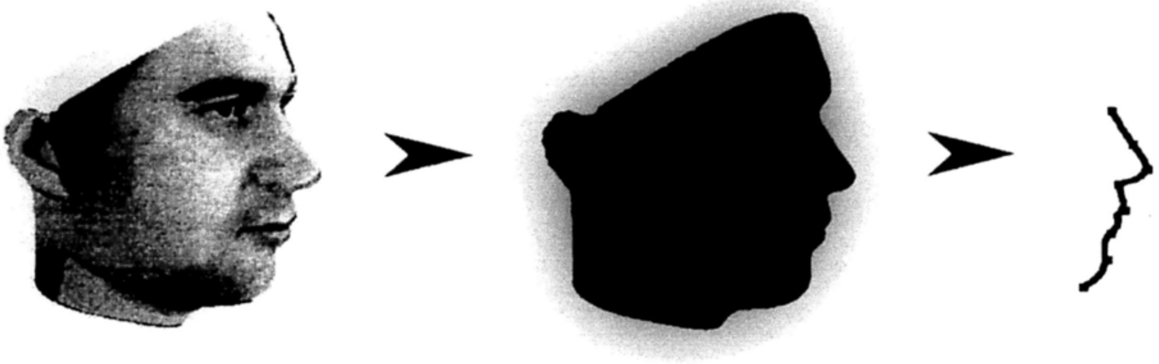


Figure 3.8: Illustration of silhouette representations by sparse points

each sample among these pixels, seeking into the mesh map for the indexes of the model vertexes on the contour.

A 3DMM does not model the entire head model. As a result, some parts of its contour, on the lower part of the neck and on the top of the forehead, are artificial. These artificial contour points are skipped at the sampling step: since the artificial part of a contour is present always in the same area of the face, a list of the vertex indexes on the artificial contour can be made so that the contour vertex indexes that are on the artificial contour list are removed. This post-processing step makes sure the validation of the contour vertex used for fitting.

Although some previous work uses silhouette information [47] for 3D motion estimation, the silhouette is often used in curve form and evolved using level-set techniques [11]. The cost of curve optimization is usually high and not suitable for real-time applications. Our point based silhouette representation is more flexible and can be efficiently integrated into a 3D pose and deformation estimation algorithm as described in Section. 3.5. Furthermore, the appearance model of the silhouette can be trained to achieve more stable performance during curve searching.

3.4.3 Type III Local Features

Point detectors are used to find interest points in images which have an expressive textures.It has been a long history that most motion estimation and tracking problems utilize the point detectors for cues extraction. A desirable quality of an interest point is its invariance to changes in illumination and homography. In the literature, commonly used interest point detectors include Harris point detector [40], KLT detector [76], SIFT [62] etc. A comparative evaluation of interest point detectors can be referred to the survey by Mikolajczyk and Schmid [67]. In the proposed tracking system, we take the KLT point detectors to find the interest points, without loss generality and performance.

Interest Point Selection

The KLT detector computes the first order image derivatives, (Ix, Iy), in x and y directions to highlight the directional intensity variations, then a second moment matrix M, which encodes the mentioned variation , is computed for each pixel in a pre-assigned neighborhood:

$$M = \begin{pmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{pmatrix} \tag{3.6}$$

An interest point is identified by finding the minimum eigenvalue $\lambda_{min}$of M:

$$\lambda_{min} = \min\{\lambda | Mv = \lambda v\}, ||v||^2 = 1. \tag{3.7}$$

The interest points are marked by thresholding λ_{min} after applying nonmaxima suppression (NMS). Similar idea is used in Harris point detector except that a new threshold rule definition R:$R = det(M) - ktr(M)^2$. The only difference between these two point detectors is that KLT enforces a predefined spatial distance between the detected interest points.

Integral Image Technique for Fast Computation If the calculation in Eq. (3.6) is implemented in a straightforward way, the computational complexity is $O(MNmn)$,where MN is the size of input image and mn is the size of a local window. This computational effort is not acceptable for most practical applications, especially when the real time tracking is concerned. Here we utilize the integral image technique to reduces the computational complexity of the traditional normalized cross correlation from $O(MNmn)$ to $O(MN)$. The proposed technique is invariant to the window size, and results in significant savings of computation time.

Given an image input $I(x, y)$ of size $M \times N$,the integral image associated with $I(x, y)$ is constructed by:

$$S(x,y) = \begin{cases} I(x,y) + S(x-1,y) + S(x,y-1) - S(x-1,y-1), & x \geq 0, y \geq 0 \\ 0, & x < 0, y < 0. \end{cases} \tag{3.8}$$

The sum of $I(x, y)$ can then be calculated from the integral image as:

$$\begin{aligned} \sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I(x+i, y+j) =& S(x+m/2, y+n/2) \\ & - S(x-m/2-1, y+n/2) \\ & - S(x+m/2, y-n/2-1) \\ & + S(x-m/2-1, y-n/2-1). \end{aligned} \tag{3.9}$$

Point Tracking

After all interest points are located, their features can be tracked across the frame by the template matching criterion. To decide whether features are being tracked successfully or not, we could examine the value of the discrepancy between the intensities of the image patch centered at the interest points across frames. However, such discrepancy function does not compensate for differences in the intensity variation among the patches of interest. A patch with high variation gives high residual because of pixel quantization and interpolation during the matching. A suitable discrepancy function turns out to be the normalized cross-correlation.

The normalized cross correlation used for finding matches of a reference template $t(i, j)$ of size $m \times n$ in an image $I(x, y)$ of size $M \times N$ is defined as:

$$N(x,y) = \frac{\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} t(i+m/2, j+n/2) I(x+i, y+j) - mn\mu_I \mu_t}{\left(\sum_{i=-m/2}^{m/2} \sum_{j=-n/2}^{n/2} I^2(x+i, y+j) - mn\mu_I^2 \right)^{\frac{1}{2}} \left(\sum_{i=0}^{m} \sum_{j=0}^{n} t^2(i,j) - mn\mu_t^2 \right)^{\frac{1}{2}}}, \tag{3.10}$$

where $\mu_I(x, y)$ is the mean of local patch centered at (x, y) and μ_t is the mean of the reference template. In the feature tracking stage, we find an instance of a small reference template from previous image around the interest point. By sliding the template in current image by a pixel-by-pixel basis, we compute the normalized correlation between them. The maximum values or peaks of the computed correlation values indicate the matches between a template and image patch in current frame. Meanwhile, the integral technique is used here again for the computation of $I^2(x, y)$.

Since the 3D pose and deformation parameters of previous frame and key-frames are assumed to be known, the 3D correspondences of the interest points can be calculated. It should be noted that the online matching of interest points could have many outliers. With the estimate of current 3D pose and deformation parameters, we can accordingly reject the matches which violate the geometric constraints among the previous frame, current frame and key-frames. Such rejection technique is applied in the pose and deformation estimation process described in Section. 3.5.

3.5 Hierarchical Shape and Pose Inference

The 3D face parameters to be estimated are denoted as $\{\alpha, Q\}$,where $Q = \{R, T\}$ represents the rigid motion of six parameters (three for object rotation and three for its translation); α is the coefficient of the linear model in Eq. (3.1). Given the 2D-3D correspondence between current input image and 3D morphable model, the simultaneous pose and deformation recovery is an ill-posed problem with the following objective function:

$$\{\hat{\alpha}, \hat{Q}\} = arg \min_{\alpha, Q} \Big(\sum \rho \left(||\Psi_c(u, Q, \alpha)||^2 \right) + \sum \rho \left(||\Psi_s(u, Q, \alpha)||^2 \right) + \sum \rho \left(||\Psi_i(u, Q, \alpha)||^2 \right) + \beta \sum_{i=1}^{k} \alpha_i^2 / \lambda_i^2 \Big). \tag{3.11}$$

The derivation of Eq. (3.11) can be found in the appendix.

In the equation above, Ψ is an essential optimization function which denotes the 2D projection error between a 3D model point and its 2D correspondence u. The 3D points are represented by the deformation parameters α. The footnotes of Ψ_c, Ψ_s and Ψ_i represent the errors of the three types of features respectively. All types of correspondence errors are treated equally. ρ is a robust function [81] with a threshold$T$:

$$\rho(r) = \begin{cases} r & r < T \\ 2T & r \geq T \end{cases} \tag{3.12}$$

The last regularized item in Eq. (3.11) models the constraint of the deformation from 3D morphable model in Eq. (3.1). In our experiment's configuration, T indicates the sum of square projection error from 3D model.

As shown in [41], α and Q in Eq. (3.11) are coupled in the non-linear function Ψ. A two-stage optimization scheme is taken where each item in $\{\alpha, Q\}$ is fixed in turn for optimal inference. In general case, such a sequential optimization scheme often suffers from the non-trivial local optima. Hence, we

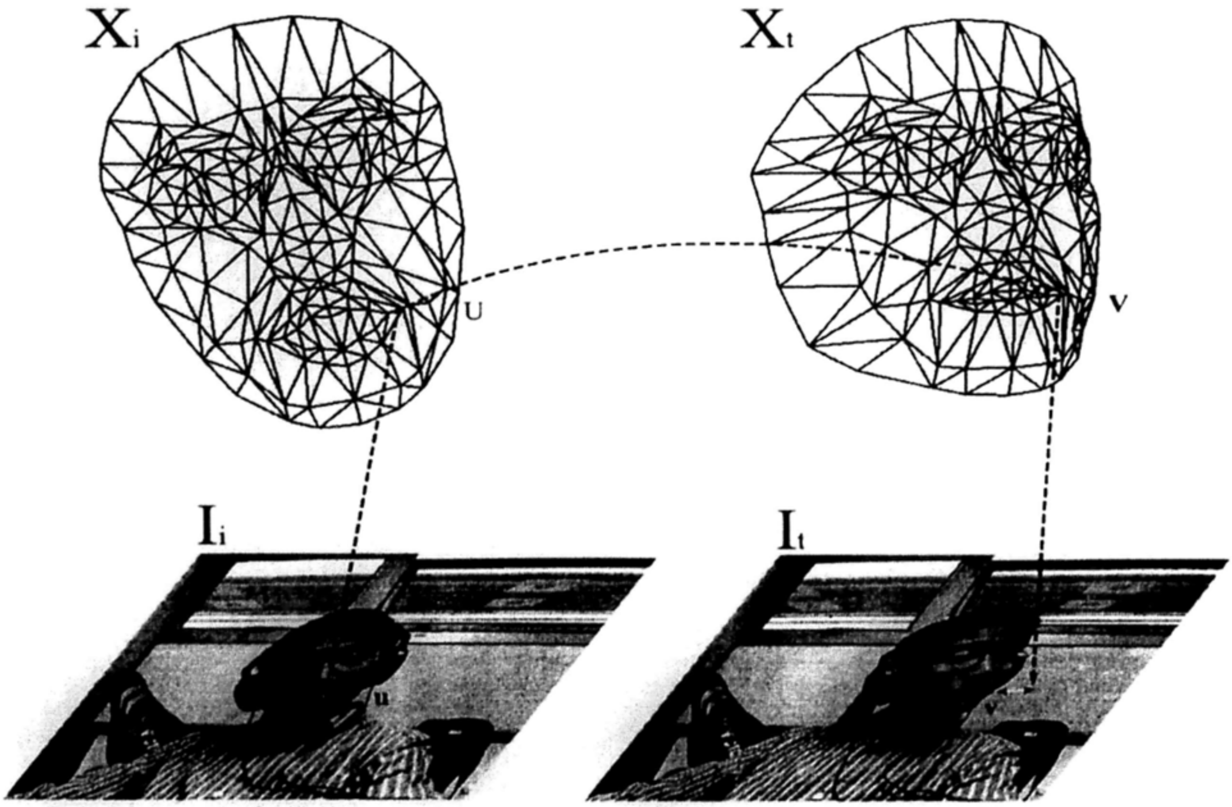


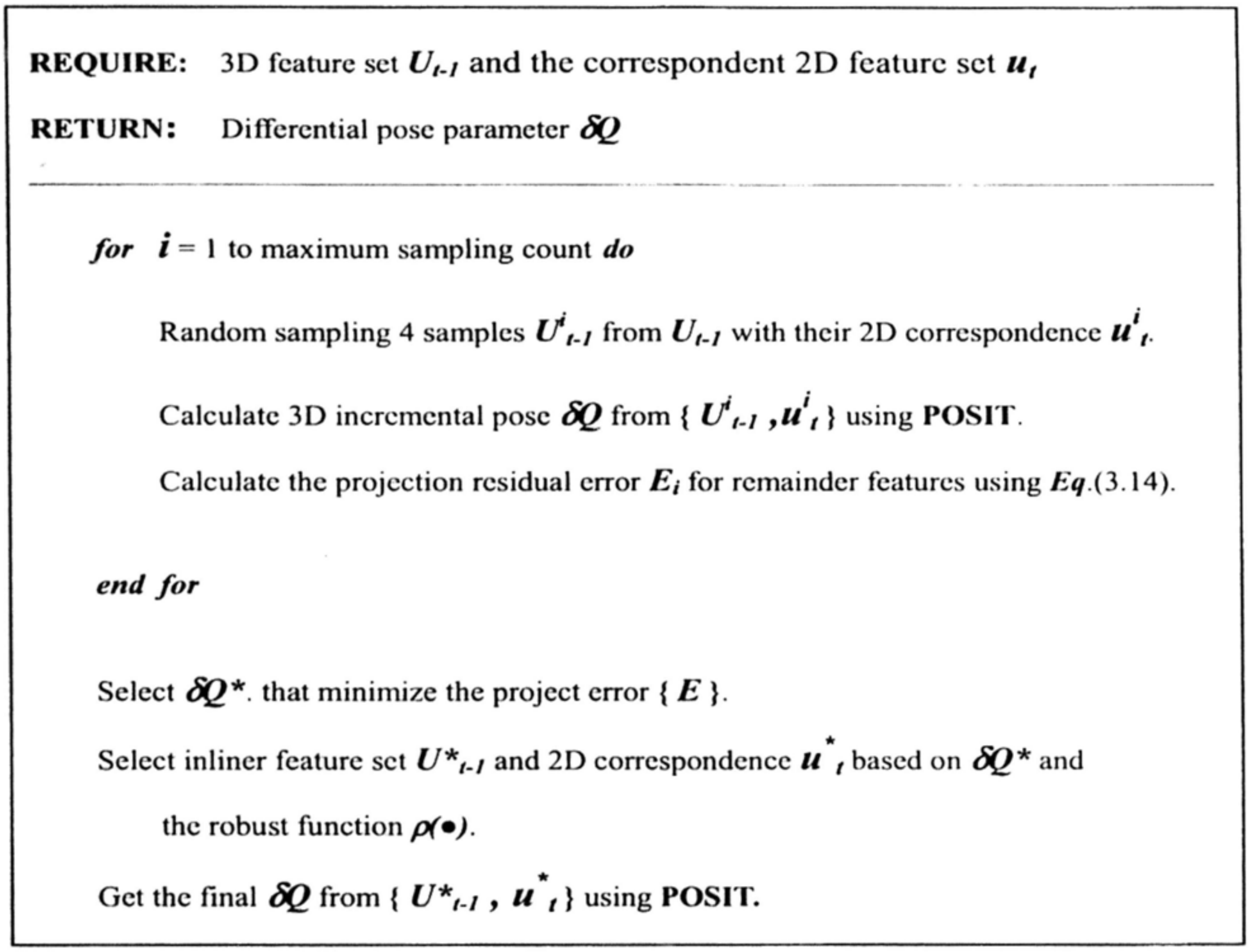
Figure 3.9: Likelihood for inter-frame motion

make use of a hierarchial optimization strategy described in Section. 3.5.2 to estimate the deformation and pose parameters efficiently.

3.5.1 Robust Initial Pose Estimation

Initial pose is estimated based on the rigid assumption for the inter-frame motion. Given the deformation parameters estimated in previous frame, the initial pose is obtained via minimization of the first item in Eq. (3.11).

The initial pose estimation for the inter-frame $\{u_{t-1}, u_t\}$ is defined based on the point matches between frame $t-1$ and t,as shown in Figure.3.9. Given the 2-D points set u_{t-1} in frame $t-1$ and its back-projection 3-D points U_{t-1}, U_t is denoted as the 3-D point correspondence to U_{t-1} in frame t and u_t is the 2-D correspondence to u_{t-1}. From the definition, we can see that U_t is easily to obtained from U_{t-1},by applying the inter-frame motion $\delta Qt-1, t$thus we



```
REQUIRE:  3D feature set $U_{t-1}$ and the correspondent 2D feature set $u_t$
RETURN:   Differential pose parameter $\delta Q$
--------------------------------------------------------------------------------
    for $i$ = 1 to maximum sampling count do
        Random sampling 4 samples $U^i_{t-1}$ from $U_{t-1}$ with their 2D correspondence $u^i_t$.
        Calculate 3D incremental pose $\delta Q$ from { $U^i_{t-1}$ ,$u^i_t$ } using POSIT.
        Calculate the projection residual error $E_i$ for remainder features using Eq.(3.14).
    end for
    Select $\delta Q^*$. that minimize the project error { $E$ }.
    Select inliner feature set $U^*_{t-1}$ and 2D correspondence $u^*_t$ based on $\delta Q^*$ and
        the robust function $\rho(\bullet)$.
    Get the final $\delta Q$ from { $U^*_{t-1}$ , $u^*_t$ } using POSIT.
```

Figure 3.10: 3D Pose Estimation from 2D-3D Points Correspondence

define the likelihood of the inter-frame motion as follows:

$$\hat{\delta}Q_{t-1,t} = \arg\max_{\delta Q}\left(\sum_{i=1}^{m}\rho(e_{ti}^2)\right) \tag{3.13}$$

where e_{ti} is the position difference between the i-th 2-D point u_t and the image projection of 3-D point U^i_{t-1}:

$$e_{ti}^2 = ||u_t^i - T_{\delta Q_t}(U_{t-1}^i)||^2 = ||u_t^i - A[R|T]\mathbf{U}_{t-1}^i||^2 \tag{3.14}$$

The accuracy of pose estimation depends on the correctness and accuracy of the correspondences between $\mathbf{U}_{t-1}$and u_t, which is determined by the 2-D feature matches between u_{t-1} and u_t in turn.

3.5.2 Two-stage Parameters Estimation

Estimate Q Given α Here, the cost function we need to minimize is:

$$E(Q) = \rho(||\mathbf{x} - P(Q(X))||^2). \tag{3.15}$$

where $Q(X)$ is the rigid transform with rotation matrix R and translation vector T, P is the perspective projection of the 3D vector $Q(X)$ in the image plane. We employ the stochastic optimization approach again from 3.10, to optimize Eq. (3.15) as follows:

Estimate α Given Q When the pose parameter Q is fixed, the cost function in Eq. (3.11) can be re-arranged with the following form:

$$E(\alpha) = \rho(||\mathbf{x} - P(H(\alpha))||^2) + \beta \sum_{i=1}^{k} \alpha_i^2/\lambda_i^2. \tag{3.16}$$

where H denotes the linear transform of all the 3-D model points from the shape parameter α. The first item in Eq. (3.16) is still a non-linear form with a high dimensional parameter α.

Here we deduce the minimization of the first item in Eq. (3.16) to the following objective function:

$$E_s(\alpha) = ||M\alpha - b||^2, \tag{3.17}$$

where matrix M is a linear transform matrix stemming from $\{Q, \Phi, x\}$. We also prove that the solution $\hat{\alpha}$ for Eq. (3.17) is also the optimal solution for Eq. (3.16) when the linear equation $\{M\alpha = b\}$ has the unique solution; otherwise, it is the solution with the assumption that the projection degenerates to the weak-perspective model, which is the assumption used in [41],[42]. The detailed inference is list in Appendix A.

In order to obtain the optimal shape parameters, we set the partial derivative of Eq. (3.16) to zero. The optimal parameters are obtained by solving the

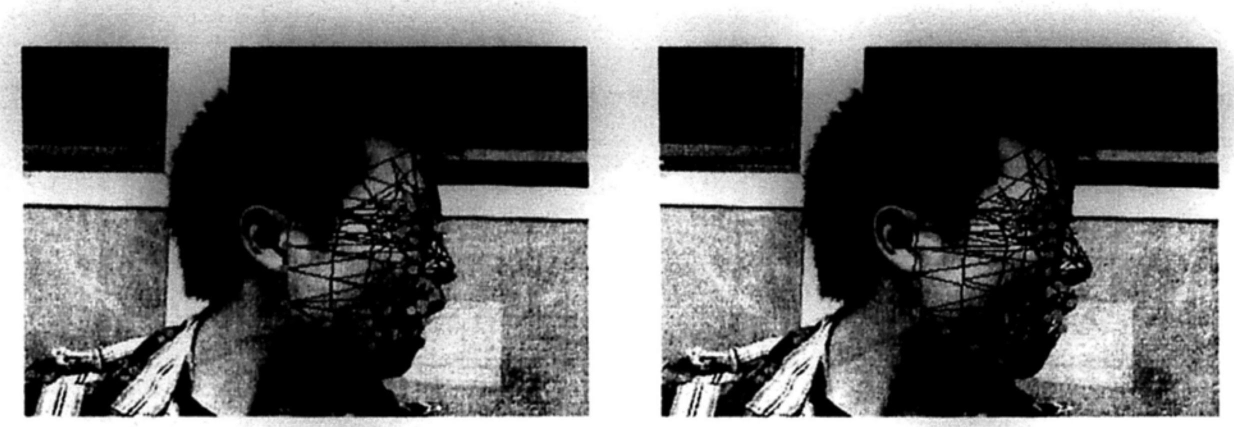


Figure 3.11: Model fitting result given an initial pose

following linear equation:

$$\left(M^T M + \beta\Lambda\right) \alpha = M^T b. \tag{3.18}$$

3.6 Experiments

3.6.1 View Based Local Descriptors

As shown in Figure. 3.2 view based local models for semantic features are constructed as follows:we render the 3D textured face models in 11 different views, where the roll angle ranges from −60° to 60° and the pitch angle ranges from −30° to 30°, with a step angle of 30°. Given the location of each feature in each view, one image patch centered at the location is extracted as positive sample and 24 neighboring image patches are extracted as negative samples. In our experiment, the local feature models are built on three levels of image pyramid. The matching of the semantic feature is carried out as follows: firstly the view of the feature model is selected by the current 3D pose estimation, then the feature match is obtained by searching in the neighborhood for the position with lowest feature model energy.

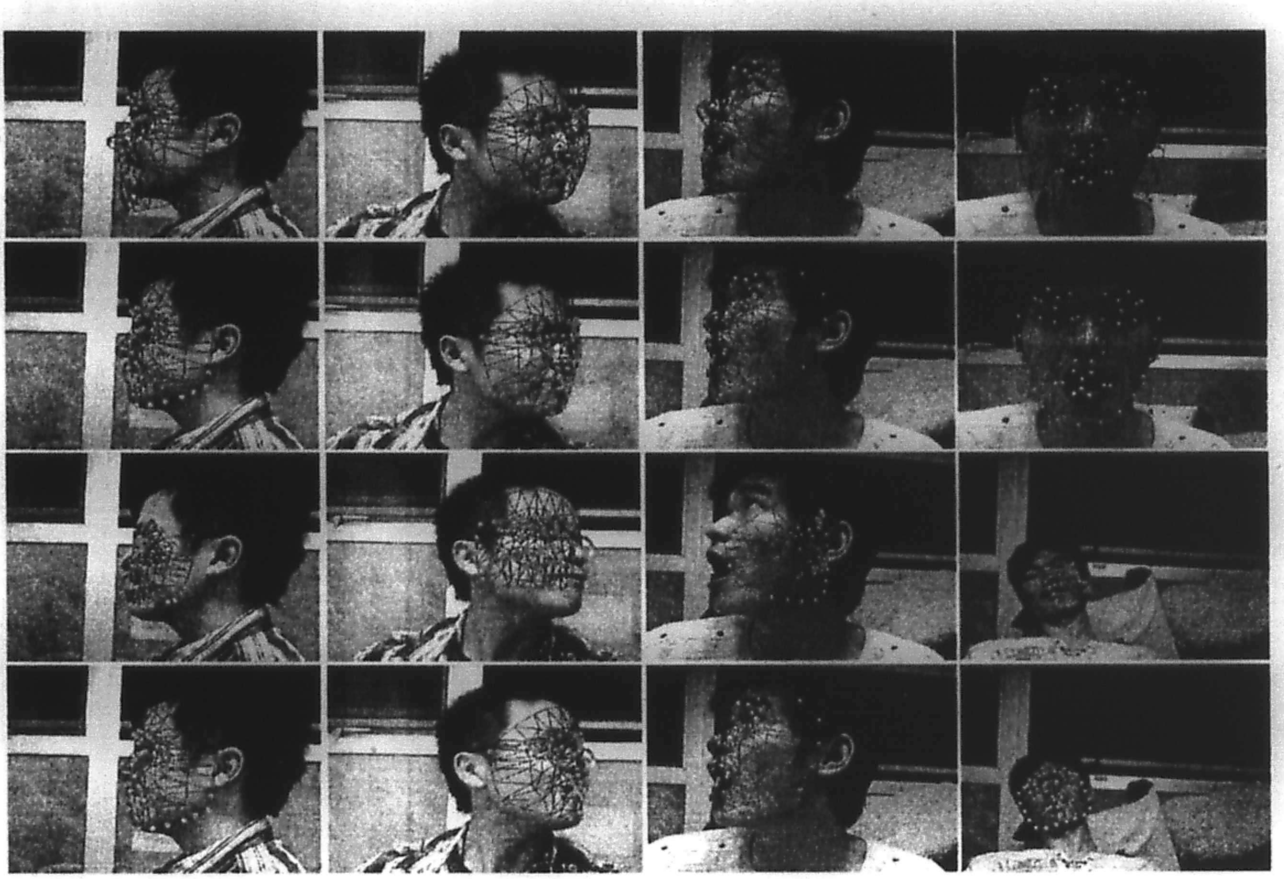


Figure 3.12: The first and third rows are the tracking results without silhouette features and interest points respectively, the corresponding results of the proposed method are listed in the second and fourth rows.

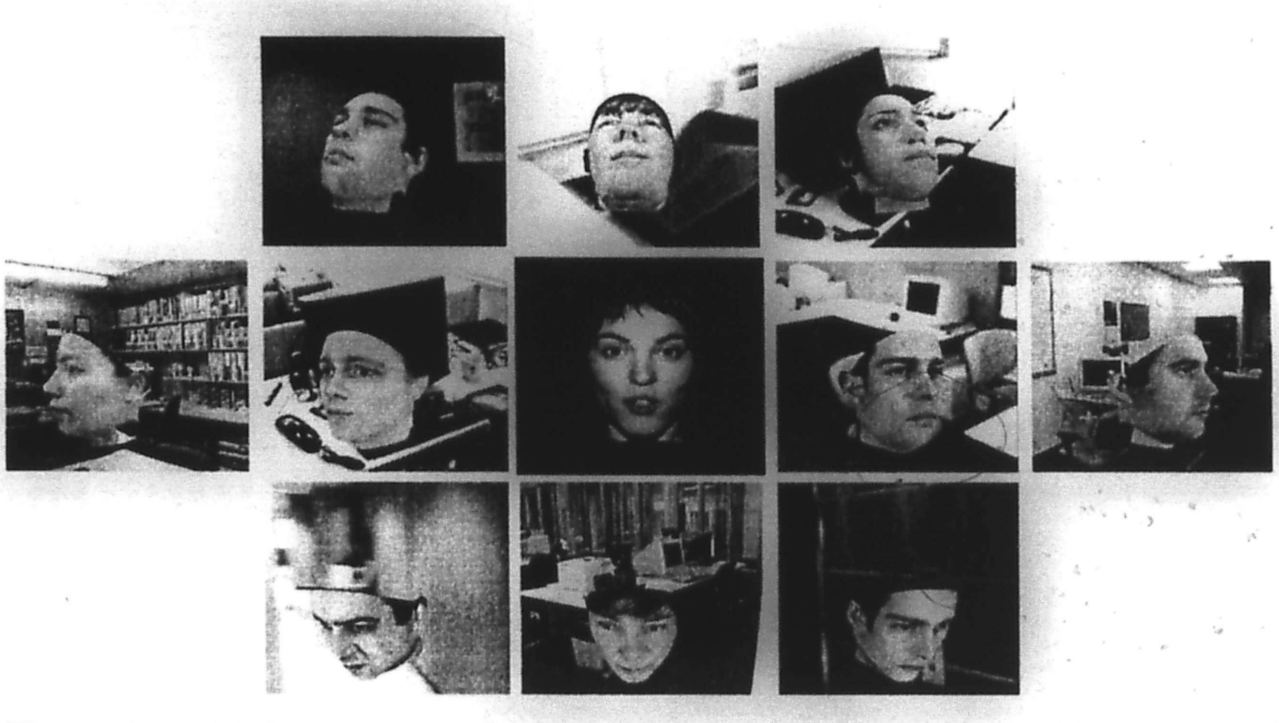


Figure 3.13: Multi-view samples for view based models training: The 11 view of training samples are collected, to enhance the feature's discrimination power on the border of facial region, we add random images from MIT scene database[82] as background for more variation.

3.6.2 Video Sequence Selection

To test the proposed algorithm on the videos in the indoor environment, we build a video database with 182 persons. Each identity is required to captured videos with similar configurations: the first 100 frames hold mainly frontal pose with neural expression; thereafter, one has around 600 frames in which no constraints are posed on the head motion and the expression variation except that the sequence starts from a frontal pose. Meanwhile, the video clips are captured in different environment and day-time. As a consequence of these freedom, the face images of the testing sequence vary greatly with the out-of-plane rotation, different facial expressions and uncontrolled illumination conditions. The typical video resolution is 640 × 480 pixels. We select 400 interest points, 68 semantic features, and 15 silhouette features for non-rigid face tracking to achieve the results in this thesis.

The non-rigid face motion are shown by projecting the 3D morphable model into the image plane under the estimated 3D pose. Meanwhile, the semantic features and silhouettes features are also imposed on the image for visualization purpose. For better tracking result illustration, we use only half of those features for demonstration. In Figure. 3.11, the initial estimated pose from interest points matching is shown in the first image, the iterative shape and pose estimation is performed and the final tracking result is illustrated in the second image. The fitting process converges to the right solution within 10 iterations.

To verify the robustness and generalization ability of the proposed tracking method, the algorithm has also been tested on the image sequences captured from a USB camera for four different persons. The difficulties for tracking in the image sequence stem from the out-of-plane motion, large expression change and numerous occlusions. Aside from providing the tracking output, a set of comparison experiments are performed to demonstrate the rationality of the integration of complementary features. More tracking results for the captured image sequences can be obtained from the supplementary materials or downloaded from [108].

In the live video experiments, the typical USB camera's resolution is set to $320 * 240$, and 100 interest points, 68 semantic features, and 15 silhouette features are used to achieve the results in Figure. 3.17. As for the key-frame setting, we use two key-frames for robust estimation: one key-frame is simply the first input frame and another key-frame is selected from a pool of history frames whose parameters are reliably estimated and are moderately similar to the current frame. The threshold value of T in Eq. (3.12) takes the fixed value 6 for all the test samples.

The tracking results in the first two rows of Figure. 3.12 demonstrate the importance of the silhouette features for face boundary localization especially in the near profile pose, where the available semantic features reduce by half with comparing to those of the frontal face. The other two types of features

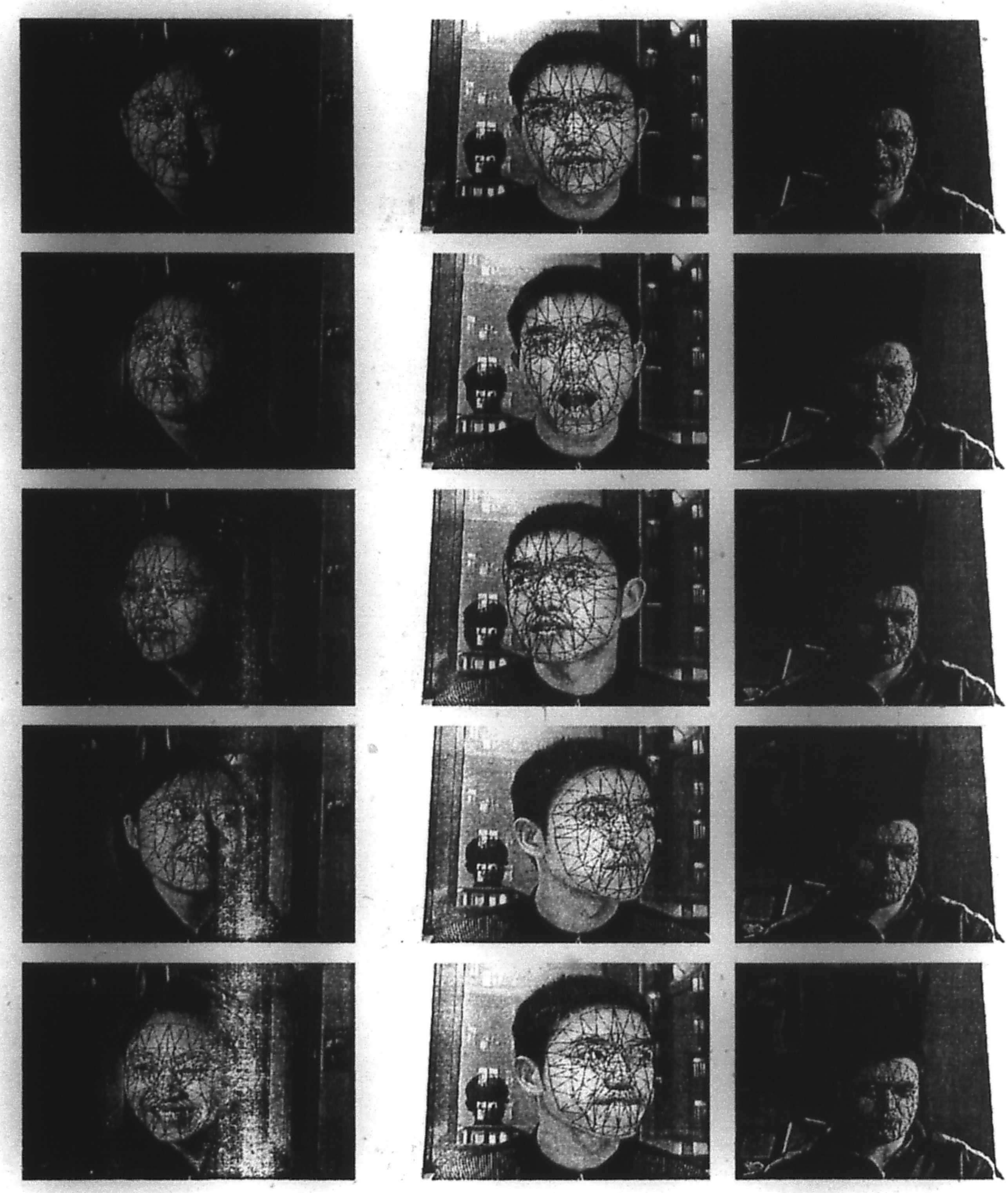


Figure 3.15: Tracking with vary illumination

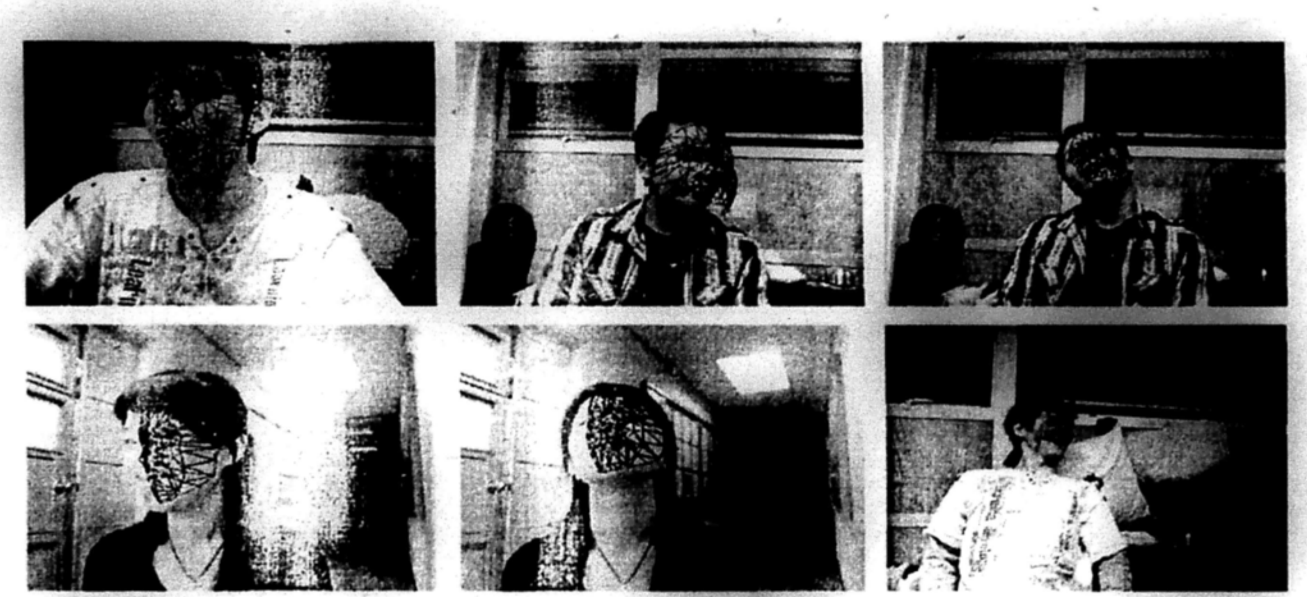


Figure 3.16: Bad tracking results with live captured image sequences. The mis-alignment of the model is mainly due to insufficient of localization power of the local features, as shown by the green points.

are kept the same for a fair comparison. The tracking results in the last two rows of Figure. 3.12 illustrate the key role of the interest points on rigid motion estimation. The tracking without the interest points suffers from agile motion where the search of semantic points cannot reach accurate positions.

Figure. 3.17,3.18,3.19,3.20 shows some typical tracking results of the selected videos. Our tracking algorithm accurately localizes the facial components such as eyes, eyebrows, noses and mouthes, under the conditions with agile motion, large expression variation, and large occlusion.

The proposed tracking algorithm runs fast on the general PC. On a Pentium-D 3.6G computer, the algorithm's speed is about 15 fps for the video with $320 * 240$ resolution. The quantity analysis of the algorithm shows that 90 percentage of the running time is spent on the type I and type II local features matching. The performance of the proposed tracking is proportional to the number of features used for offline searching.

There still exists some failure cases for the proposed tracking approach, as shown in Figure. 3.16. *When the tracking target turns to a side view pose with* *large expression variation, the proposed tracking approach is failed to track the subtle motion around the semantic facial component.* The most failure cases are due to the insufficient localization ability of the local feature models when side view faces are presented. We attribute the insufficient discrimination power of the local feature to the the synthesized training samples via 3D face reconstruction approach. The facial texture of the synthesized samples are obtained from image observation, in which the self-occlusion part is interpolated from the visible part.

3.7 Conclusion

In this chapter, we propose an efficient approach for tracking the deformable 3D face from a single camera. By taking the advantage of complementary features integration, we obtain a large number of 2D-3D correspondences for the estimation of pose and deformation parameters. Aside from the low cost of computation, the usage of local features enhances the generalization ability of the tracking system. The non-linear shape and pose optimization is efficiently solved by a hierarchical optimization scheme. The whole tracking process is efficient, automatic, and the result achieved is accurate even for low quality video.

Figure 3.17: Tracking results for the first performer. The selected frames show large pose variation and deformation.

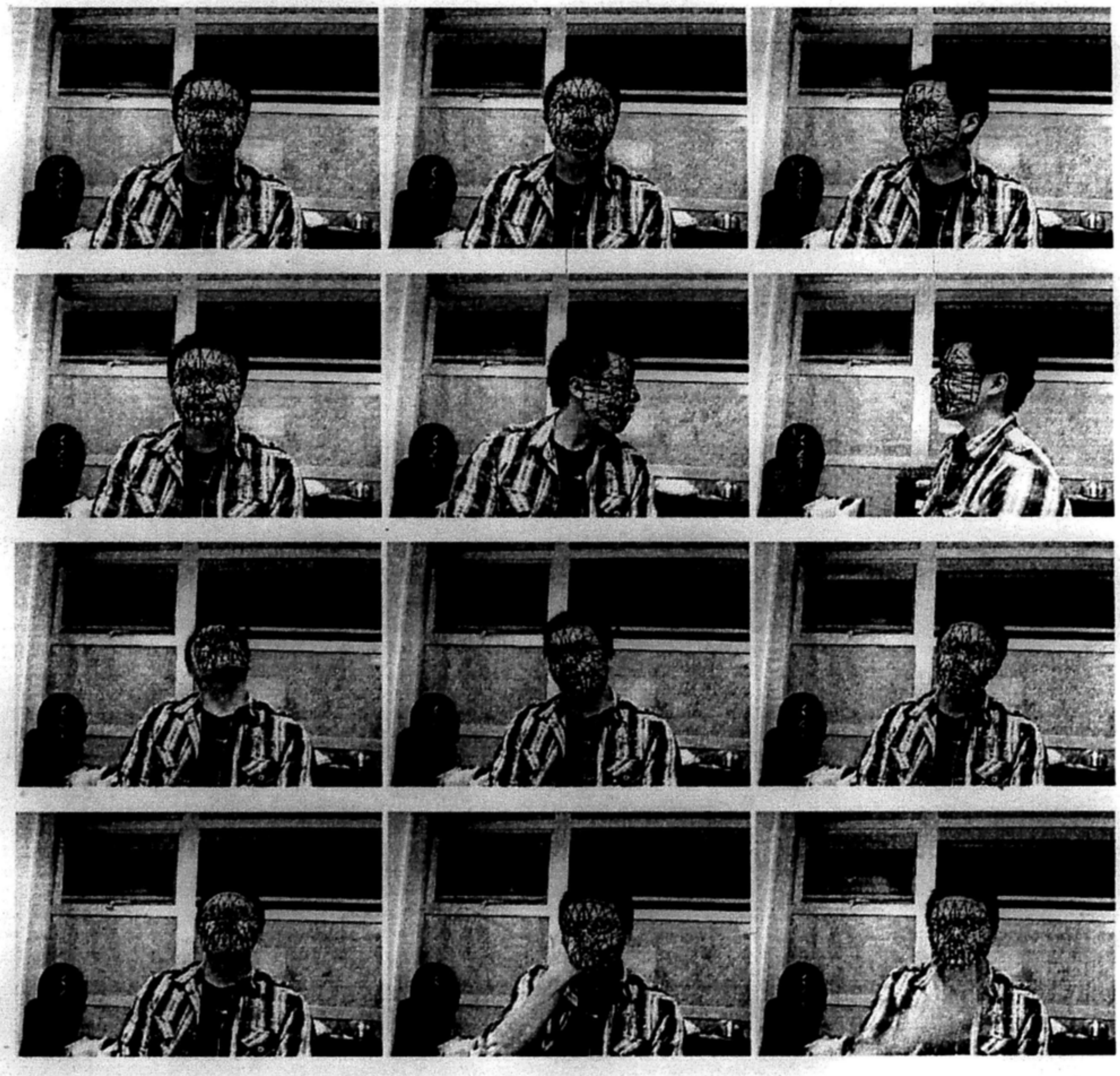


Figure 3.18: Tracking results for the second performer. The second image at the first row shows a bad fitting result on the excessive expression. This is due to the lack of extreme expression variation in our training database.

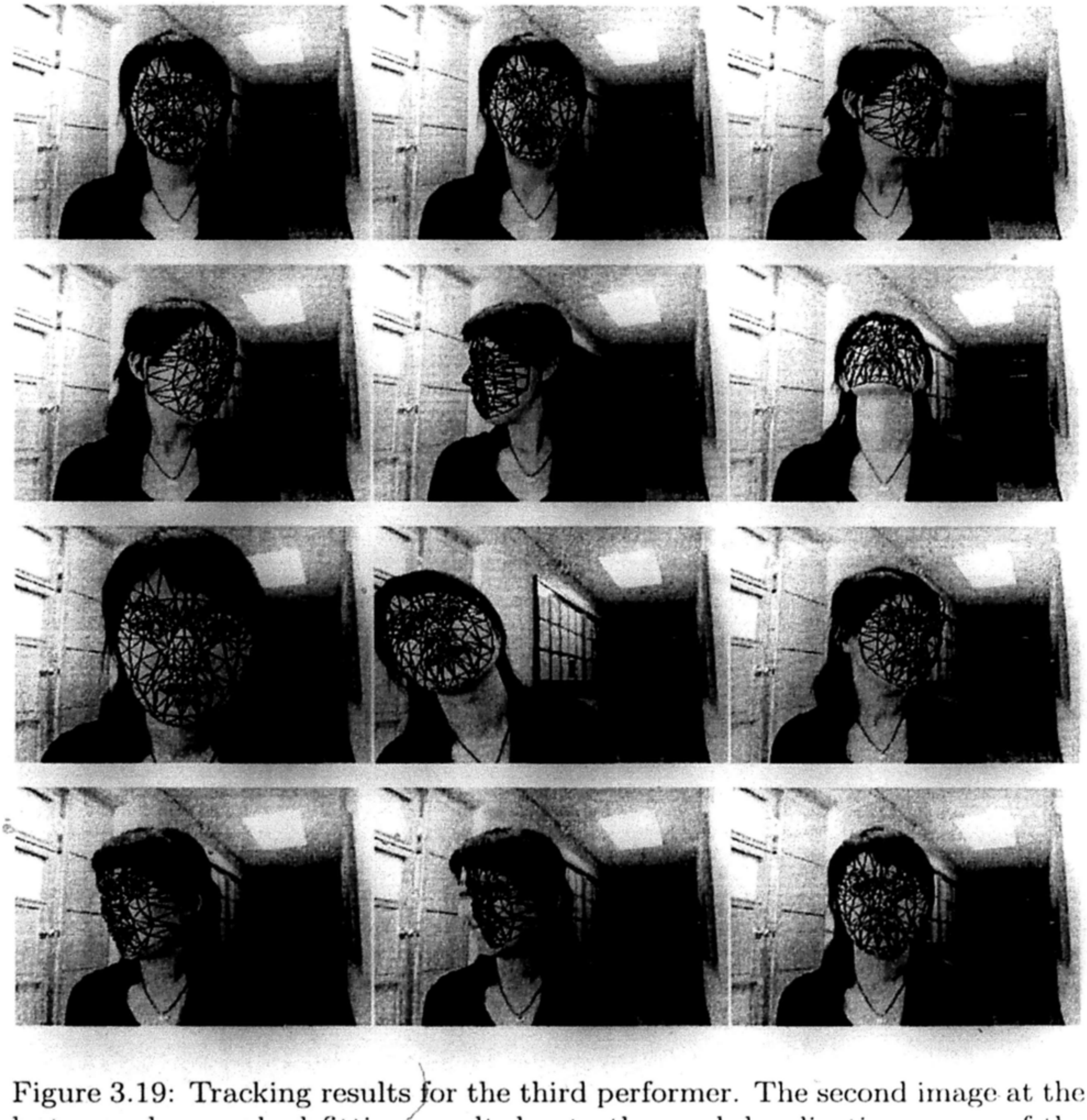


Figure 3.19: Tracking results for the third performer. The second image at the last row shows a bad fitting result due to the weak localization power of the side view feature. However, it can be quickly recovered when the near frontal pose is given, as shown in the last image.

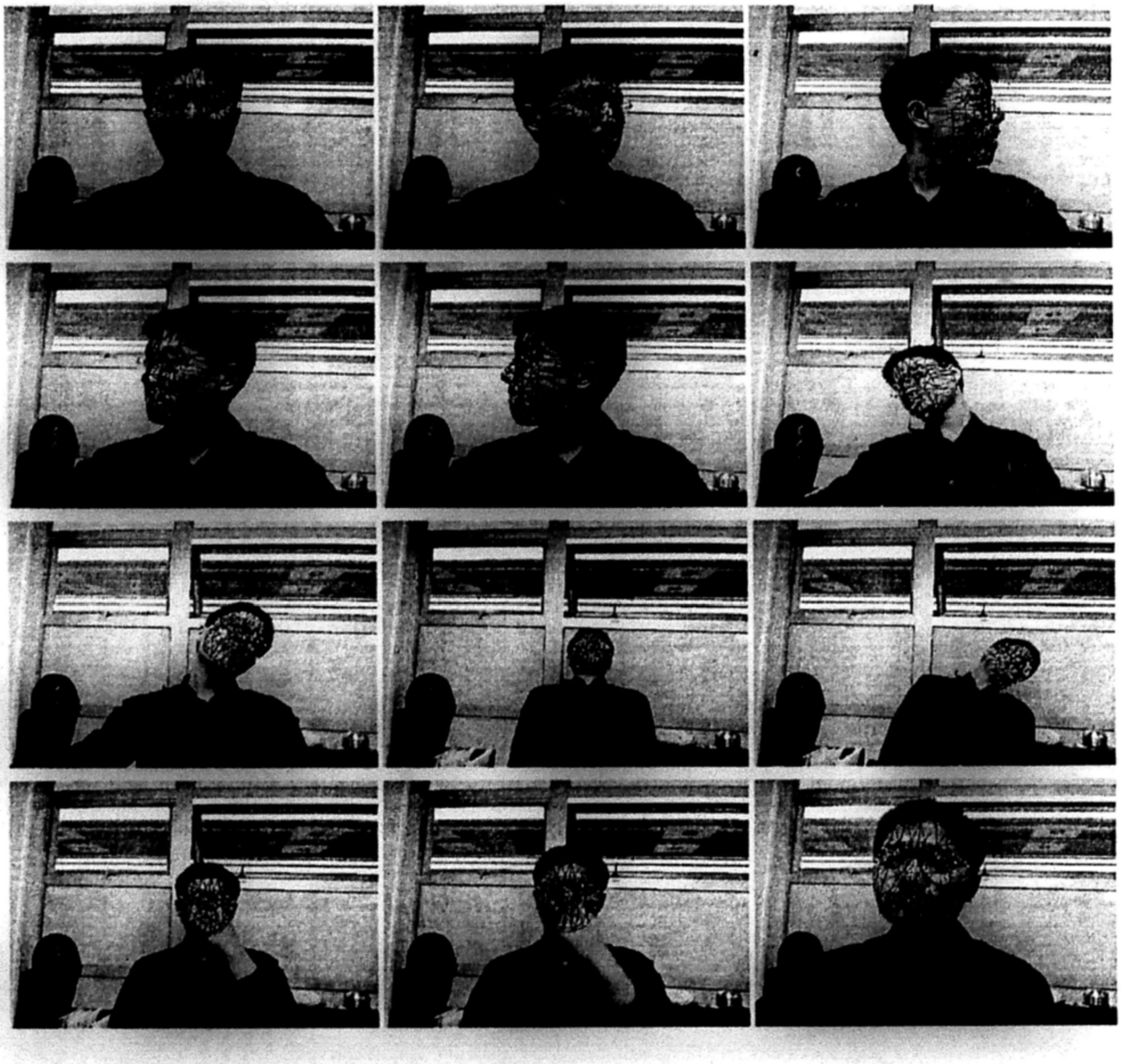


Figure 3.20: Tracking results with the forth performer. The first two images at the last row illustrate the robustness of the tracking with partial occlusion.

3.8 Appendix

3.8.1 Bayesian Interpretation of NonRigid 3D Tracking

Probabilistic formulation is frequently used for visual tracking since they provide a mathematical foundation for the derivation of target's state distribution in a dynamic system. It provides a principled framework for fusing information from independent sources, including both offline source and online observations. In this section, we provide a probabilistic interpretation to our non-rigid 3d face tracking problem, and give the detailed derivation of Eq. (3.11).

The non-rigid face tracking is modeled as a dynamic system, where the state of the target and image observation at the t-th frame are represented by X_t, I_t respectively. Given the image observation sequence $\mathbf{I} = \{I_1, \ldots, I_k\}$ that related to the inference of tracking status in current frame, the correspondent target states are denoted as $\mathbf{X} = \{X_1, \ldots, X_k\}$, tracking at frame t is to infer the posterior distribution $P(X_t|I_t, \mathbf{I}, \mathbf{X})$. α is the shape parameter that incorporate the prior of deformable 3D model.

In Figure. 3.21, the posterior distribution of X_t is specified by:

$$P(X_t|I_t, \mathbf{I}, \mathbf{X}, \alpha) \propto P(I_t, |X_t, \mathbf{X}, \alpha)\, P(X_t|\mathbf{X})\, P(\mathbf{X}|\alpha)\, P(\alpha). \tag{3.19}$$

If we assume a static prediction model $p(X_t|\mathbf{X}) \sim N(0, \delta I)$, which means the predication of the current state X_t from the related state set $\mathbf{X}$ holds unchanged. Meanwhile, the deformation prior α is modeled as a multivariate Gaussian:

$$\begin{gathered}\alpha \sim N(0, \Lambda), \\ \Lambda = diag\{\lambda_1, \lambda_2, \ldots, \lambda_N\}, \\ P(X|\alpha) \propto N(\mu + \Phi\alpha, \delta_1 I).\end{gathered} \tag{3.20}$$

The local feature matching from consecutive frames can be enforced by incorporating another Gaussian model into E_{proj} in Eq. (3.14):

$$P(I_t, |X_t, \mathbf{X}, \alpha) = \frac{1}{Z_{proj}} \prod_i \exp(E^i_{proj}/\delta_2), \tag{3.21}$$

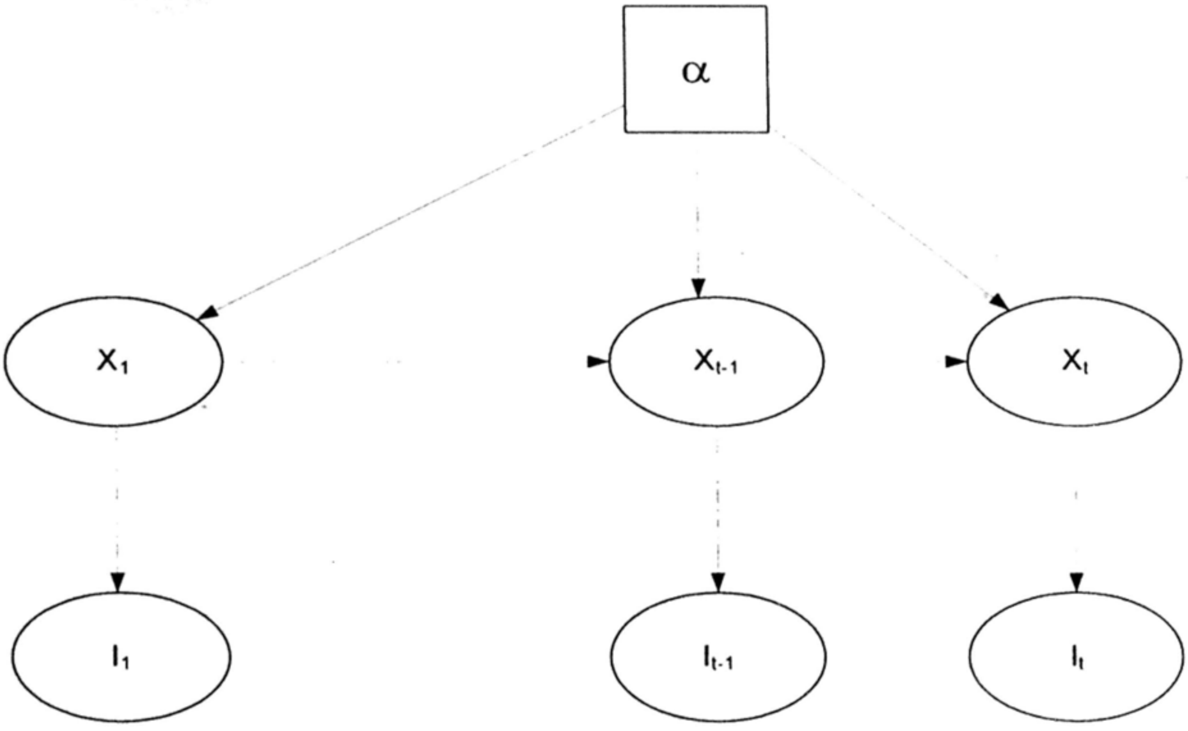


Figure 3.21: Graph Model for Non-Rigid 3D Tracking

where Z_{proj} is a normalization factor to ensure the sum to be one. The Gaussian assumption imposed in Eq. (3.20) and Eq. (3.21) guarantee the posterior $P(X_t|I_t, \mathbf{I}, \mathbf{X}, \alpha)$ is still a Gaussian. The MAP estimation on X_t can be converted to an energy minimization form

$$E(X_t) = \mathbf{k}(\alpha^T \Lambda^{-1} \alpha) + \sum_i E^i_{proj}, \tag{3.22}$$

where $\mathbf{k}$ is the inverse of the variance defined on the conditional probability distribution $p(\alpha|I_t)$ with integrating out all the intermediate variables.

3.8.2 Estimate the Shape Parameters From Fixed Pose

Given the pose parameter $Q = \{R, T\}$ fixed, the projection error E_{proj} in Eq. (3.16) can be expanded to the following form with the perspective projection assumption:

$$\sum_{i=1}^{n} \left\| f \frac{\sum_1^k \alpha_j \phi'_{3i,j} + t_{i,0}}{\sum_1^k \alpha_j \Phi'_{3i+2,j} + t_{i,2}} - u_{2i} \right\|^2 + \left\| f \frac{\sum_1^k \alpha_j \Phi'_{3i+1,j} + t_{i,1}}{\sum_1^k \alpha_j \Phi'_{3i+2,j} + t_{i,2}} - u_{2i+1} \right\|^2 \tag{3.23}$$

where $T = \{t_0, t_1, t_2\}$ and $\Phi' = \{\phi_{i,j}\}$ is obtained by applying rotation matrix R to the eigen-vectors Φ of shape PCA:

$$\Phi' = \begin{pmatrix} R & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & R \end{pmatrix} \Phi. \tag{3.24}$$

Consider the following the equation:

$$f \frac{\sum_1^k \alpha_j \phi'_{3i+l,j} + t_{i,l}}{\sum_1^k \alpha_j \Phi'_{3i+2,j} + t_{i,2}} = u_{2i+l}, l \in \{0, 1\}. \tag{3.25}$$

Eq. (3.25) can be re-arranged in a linear form as:

$$M_{2n,k} \alpha = \mathbf{b}_{2n} \tag{3.26}$$

where

$$\begin{aligned} m_{2i+l,j} &= f\phi'_{3i+l,j} - u_{2i+l}\phi'_{3i+2,j}, \\ b_{2i+l} &= u_{2i+l}t_2 - ft_{i,l} \end{aligned} \tag{3.27}$$

In the remainder part, we will prove that: **If Eq.** (3.26) **has the unique solution, i.e.,**$Rank(M) = Rank(M|b)$**, the original function in Eq.** (3.23) **also reaches its optimal value; otherwise, the solution** $\hat{\Lambda}$ **for Eq.** (3.26) **is the optimal solution in the meaning of MSE(minimum square error), with the first order approximation of the projection model.**

The residual error ϵ of the linear equation Eq. (3.26) is denoted as

$$\epsilon = (M\hat{\Lambda} - b)^T (M\hat{\Lambda} - b). \tag{3.28}$$

Therefore, E_{proj} in Eq. (3.23) can be expressed by ϵ as following:

$$E_{proj} = (M\hat{\Lambda} - b)^T \begin{pmatrix} \frac{1}{Z_1^2} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{1}{Z_1^2} & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \frac{1}{Z_n^2} & \vdots \\ 0 & 0 & \cdots & \cdots & \frac{1}{Z_n^2} \end{pmatrix} (M\hat{\Lambda} - b), \tag{3.29}$$

where Z_i is the depth of i-th vertex on the 3D model after applying the rigid transformation Q:

$$Z_i = \sum_{1}^{k} \alpha_j \Phi'_{3i+2,j} + t_{i,2} \tag{3.30}$$

If $\{t_{min}, t_{max}\}$ are denoted as the minimum/maximum depth value of all vertices, the following inequality holds:

$$\frac{1}{t_{max}^2}\epsilon \leq E_{proj} \leq \frac{1}{t_{min}^2}\epsilon \tag{3.31}$$

A small ϵ means a narrow range for the optimal value E_{proj}. Under the assumption of weak-perspective projection model, in which t_{max} equals t_{min} for all the points, the optimal values is equivalent in Eq. (3.23) and in Eq. (3.25), which means the solution in Eq. (3.25) is the solution with the weak perspective model assumption in the worst case.

Chapter 4

Video Based Face Recognition

In this chapter, we propose a framework for the task of face recognition in real-world noisy videos based on 3D deformable face tracking. The difficulty of video face recognition tasks lies in the challenging appearance variations in real-world videos due to motion blur, large head rotation, occlusion, illumination change and significant image noise. To achieve the goal of accurate localization of faces in videos, the proposed 3D tracking algorithm makes good use of 3D face shape priors, local appearance model of major facial features, face silhouette and online feature matches across video frames. As opposed to the state-of-the-art video face recognition algorithm which relies on discriminative appearance model to classify face images into different views, our 3D tracking algorithm can directly estimate face pose for a view-based face recognition algorithm, which is more robust to outliers with comparison to the clustering based strategy. Furthermore, the proposed 3D tracking algorithm has a probabilistic form and can provide confidence measurement for the tracking result, which can be used to select high confidence frames. Such a frame selection strategy ensures the robustness of face recognition process.

With the proposed recognition framework, a high recognition rate can be achieved even with simple feature descriptors and classifiers (Regularized LDA +NN). The experiments performed on the real world noisy videos from YouTube demonstrate the significant improvement achieved with our approach:

the accurate recognition rate reaches 79.8%, while outperform the best reported results set (71.24%) on the same data.

4.1 Introduction

During the past decades, the research topics on object recognition especially the biometric related recognition such as face are extensive developed. Despite recent efforts [114, 113, 74], accurate and robust face recognition remains a challenging task in the real life applications, where the major difficulties lie in the appearance variations arising from the change in pose, expression, illumination ,partial occlusion and motion blur. At the same time, the video based face recognition has received significant attention in the field of computer vision since it can take full advantage of the temporal coherence among consecutive frames to achieve a potentially more accurate recognition rate, where the weak face observation in each frame can be integrated over the sequence.

4.1.1 Recognition: From Images To Videos

Generally speaking, all the conventional image based recognition approaches suffer from the small sample size problem in statistics and pattern recognition. For example, eigenface[84] and fisherface[72] will suffer serious performance drop or even fail to work. The core component of appearance-based recognition methods is their learning mechanisms, while the classic families of learning mechanisms (classifiers) need sufficiently large training set to achieve a good recognition performance. Such a requirement for so large a training data set is partly due to the high-dimensional representation of face images, and partly due to the appearance variation of facial expression and pose. Most of the face recognition approaches [4][55] assume several training samples of the same person are always available for training. Unfortunately, in many real-world applications, the number of training samples available is actually smaller than the requirement. Due to its challenge and significance for real-world scenarios, many techniques have been developed to attack this problem, such as virtual samples synthesis [87],[42], probabilistic patch based matching [64], neural networks approach[51], and the hybrid approach[57]. However, most of the approaches mention before focus on the task of object recognition. In the real world applications, they need a face detection and feature extraction as a preprocessing step to work automatically.

The video based approaches, from other hand, tackle the problem of small samples problem since a bunch of images, which capture the expression, pose and illumination variation, are available for one person. As shown in Figure. 4.2, a typical video-based face recognition system automatically detects face regions, extracts features from the video, and recognizes facial identity if a face is present. Face recognition based on video is preferable over using still images, since as demonstrated by Knight and Johnston[49], motion helps in recognition of faces when the images are negated, inverted or threshold. It was also demonstrated that humans can recognize animated faces better than randomly rearranged images from the same set.

Even since recognition of faces from video sequence can be directly extended from the still image based approaches, significant challenges for video-based recognition still exists:

1. Compared to the still image captured by high resolution cameras, the image quality of video frame is low. Generally, the video acquisition of face are obtained in un-cooperative style, there may be large illumination and pose variations in the face images.

2. The face size in video frame is often smaller than the assumed sizes in most still-image-based face recognition systems. For example, the valid facial region can be as small as 10 × 10 pixels (the minimum size of a detection zone), whereas the face image sizes used in feature-based

still image based systems can be as large as 128 × 128. Small-size face images not only make the recognition task more difficult since most discriminative details are lost, but also affect the accuracy of facial feature extraction, as well as the accurate localization of the fiducial points that are often needed as an preprocessing step in the recognition methods.

3. The diversity of the face images is large since large pose variation and expression variation exists. As shown in Figure. 4.6, the pose difference in the same video could be more than 60 degree. Recognition of individuals of large pose is difficult. It is still an active research area to achieve pose invariant face recognition system since the variations of the appearance caused by the view point is sometimes larger than the intra-person distance. Meanwhile, it still contains large expression variations.

4.1.2 Techniques Related to Video Based Recognition

Before we introduce the existing video-based face recognition algorithms, we briefly review the close related techniques that are important for the success of video based face recognition. In [16], four related techniques were mentioned as being important for video-based face recognition: segmentation of moving objects(faces) from video sequences; structure estimation; 3D models for faces; and non-rigid motion analysis. Based on the current development of video based face recognition, we will introduce two specific face-related techniques, facial pose estimation and face tracking, instead of the above four general criteria. These techniques are critical for the realization of the full potential of the video-based face recognition. Meanwhile, these related techniques have not been well addressed in the scenario of face recognition and still remain as the hot research fields of the video based face recognition.

Pose Estimation

It is not surprising that the performance of face recognition systems drops significantly when large pose variations exist in the input images.It has been testified in the FERET [73] and FRVT reports [74], and been proposed as a major research topic. Here the pose problem only refers to the out-of-plane rotation since the in-plane rotation can be aligned by 2D geometrical transformation.

Early methods focused on constructing pose invariant features [95] or synthesizing a fixed view image after the 3D face model is extracted from the input samples [8][42]. Such methods work well for small rotation angles and always failed when the view angle is large(60°), since severe self-occlusion cases exist. Most methods are proposed to use a large number of multiview samples to handle the large pose variations.

To address the pose problem more systematically, an attempt has been proposed by [53][48] to treat the pose estimation as a classification problem. The basic idea of this analysis is to assign a input sample with a pose label by minimize the pose distance from an image set with pose known, or by seeking the most discriminative projection direction from the pose labeled samples. More specifically, a generative pose manifold is built up by a set of local linear pose subspace. For an new input sample, the pose label is assigned to the subspace which minimize the projection error. Similar ideas can be used to train a pose discriminative subspace via LDA. The drawback of using pose classification is the large number of pose labeled samples for model training, which is hard to acquired in the real world applications.

Facial Feature Localization

The importance of facial features localization for face recognition cannot be overstated. Many face recognition systems need facial point descriptors in addition to the holistic face, as suggested by studies in psychology. It is well known that even holistic matching methods, for example, eigenfaces[84] and fisherfaces[72], need accurate key facial feature localization such as eyes, nose, and mouth to normalize the tracked/detected faces. In general, the facial feature extraction approaches can be divided into three categories:

1. general methods based on edges, lines and curves;

2. feature template based approaches that are used to detect facial features such as eyes;

3. structural matching methods that incorporate geometrical constraints on the facial features.

Early approaches focus on the localization of individual features; for example, a template matching approach was described in [39] to detect and localize the human eye in a frontal face. However, when the appearance of the features change significantly,these methods have difficulty in accurate localization. To detect these features more reliably, recent approaches have used structural matching methods, i.e, the Active Shape Model(ASM)[21]. Compared with previous approaches, the statistical based methods are much more robust in handling variations in image intensity and global shapes.

4.1.3 Existing Work

Due to the maturity of the 2D tracking techniques developed over the past decades, the majority of the existing video face recognition approaches are based on 2D face tracking [2, 48]. For the state-of-the-art approaches with the 2D face tracking schemes [48, 77], the appearance change caused by face pose or viewpoint must be learned by SVM[13], LDA[64], GMM[24] or a combination of such techniques. Moreover, the methods based on 2D tracking are limited by the rectangle based tracking results, which cannot guarantee to provide well aligned faces. Although some techniques can be developed to reject bad 2D tracking result based on face appearance learning [48], they also increase the risk to reject good tracking results. Furthermore, view classification and appearance learning significantly depend on the training data set and do not have good generalization capability for real-world applications.

There are also some 3D face tracking based approaches which try to recover rigid facial pose for recognition. In [31], a person-specific textured 3D face model is used for face pose estimation across video frames. Promising results have been achieved using a simple confidence measure for character recognition in feature movies. Similar 3D face model is used in [104] where the 3D face tracking is performed with illumination compensation. The main problem of the above approaches is that considerable amount of user supervision is required to fit a textured 3D face model for tracking and more importantly, the use of holistic appearance template or model for face tracking does not work robustly in practice.

In contrast with existing work, our tracking algorithm is based on 3D non-rigid face representation and use sparse local features as observation which include both off-line trained major facial features and online tracked image features. The resulting tracker is robust and provides accurately cropped images for view-based face recognition. The confidence of the tracking result is measured through the geometric consistency of facial features under the 3D face shape prior, which can be used to improve the face recognition performance over a sequence of frames. Through extensive experiments on the public dataset, we show that our 3D non-rigid tracking based recognition approach achieves significantly better performance over existing approaches.

The rest of the chapter is organized as follows. In Section. 4.2, the 3D tracking module is introduced, which accurately extracts localized face from video streams. At the following Section. 4.3, the features and distance measurement used for recognition is presented. In Section. 4.4, we demonstrate the performance of the proposed framework, also with the comparison with the state-of-the-art approaches. Section. 4.5 concludes the chapter and highlights the future work.

4.2 Face Tracking in 3D

In this section, we utilize the non-rigid 3D face tracking algorithm for facial information extraction. Aside from the capability of catching up the rigid 3D head motion, it can recover the non-rigid facial structure precisely across frames, which makes the proposed tracking to be able to provide well aligned samples for the following recognition task.

Given a collection of image observations $\{\mathbb{I}\}$, the non-rigid 3D face tracking at frame t is formulated as Maximum A Posterior (MAP) estimation in the Bayesian framework where the tracking state is described by the 3D shape parameter X_t and rigid pose Q_t:

$$\begin{aligned} \{\hat{X}_t, \hat{Q}_t\} &= \arg\max_{X_t, Q_t} p(X_t, Q_t | \{\mathbb{I}\}) \\ &\propto p(\{\mathbb{I}\}|X_t, Q_t) \cdot p(X_t). \end{aligned} \tag{4.1}$$

In the above equation, $p(X_t)$ is the prior distribution of 3D face shape X_t, $p(\{\mathbb{I}\}|X_t, Q_t)$ is the likelihood distribution which describes the conditional probability of image observations $\{\mathbb{I}\}$ given the tracking state $\{X_t, Q_t\}$. In the rest of the chapter the subscript t is omitted for simplification.

The proposed tracking algorithm has the following three key components:

1. the shape prior model $p(X)$, which is modeled as a deformable model and can be trained by PCA on a set of 3D shape samples;

2. the likelihood model $p(\{\mathbb{I}\}|X, Q)$, which is modeled based on the 2D-3D feature correspondences. We use an extended feature set similar to the work [110], which includes the off-line trained semantic features, face silhouette features and online tracked image features;

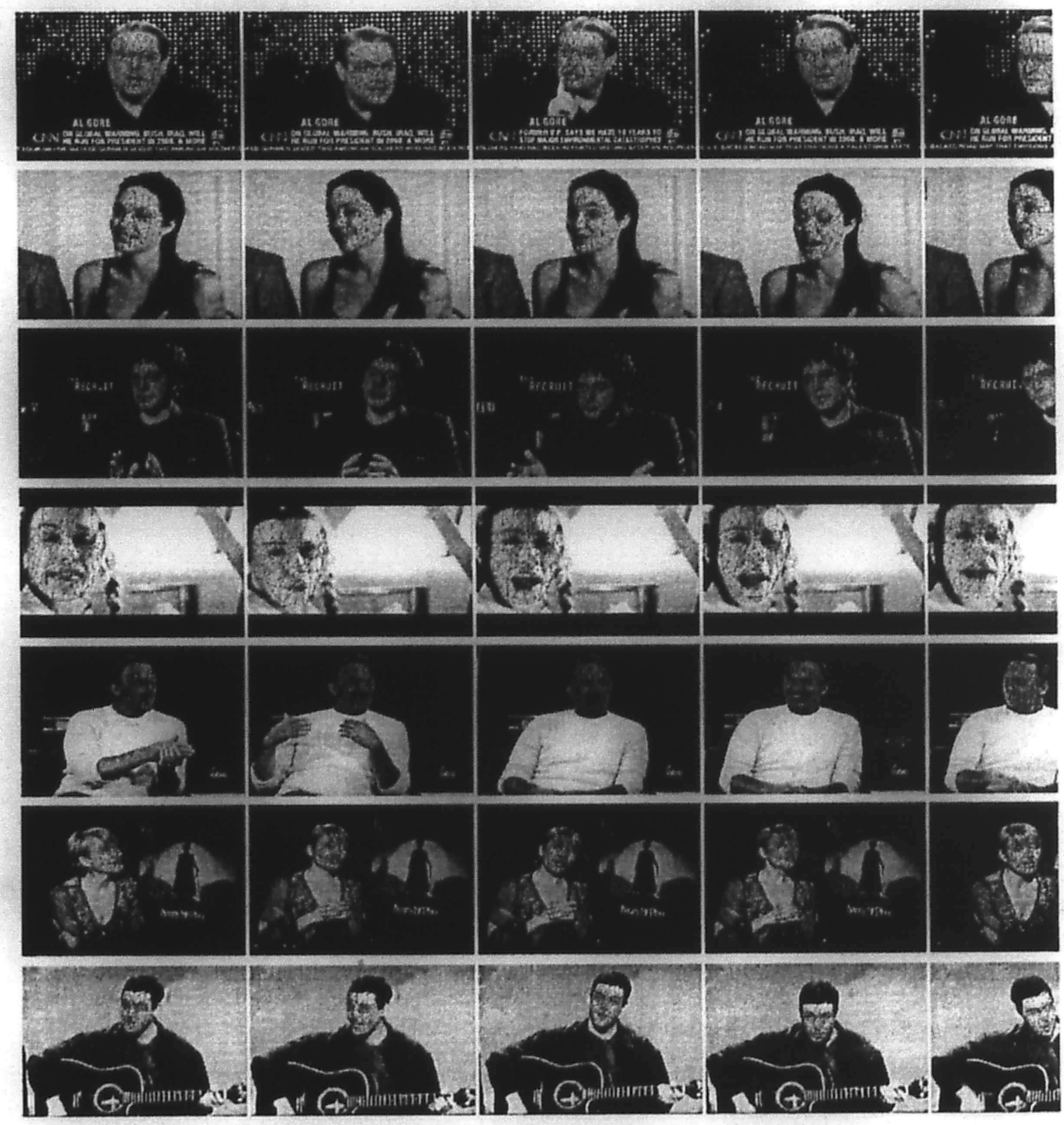


Figure 4.1: Tracking Results on YouTube

3. the robust estimation algorithm which obtains reliable results in the case of significantly noisy image observations, we make use of an hierarchical optimization strategy coupled with robust estimation techniques to fulfill the task.

The detail description about the three components is in Chapter. 3. The proposed 3D tracking algorithm has strong generalization capability, which is less sensitive to illumination, independent of tracking target. As illustrated in Figure. 4.1, the proposed non-rigid tracking approach performs well in the context of real-world applications, even with large pose, expression and illumination variation.

4.2.1 High Confidence Frames Selection

Following the output at the tracking stage, we extract faces from videos based on their 3D motion and shape parameters. However, not all the frames from a video clip are suitable for the task of face recognition since the proposed tracking module could fail on some frames even with the robust recovery mechanism. Using error tracked frames for face recognition would lead to a significant performance drop. For the purpose of robust recognition, we need to select well tracked frames as candidates frames for recognition.

The quality of the tracking result is specified by a confidence measure related to the posterior probability $p(X, Q|\{I\})$ in Eq. (3.11):

$$w = \exp\{-\frac{1}{N}\sum_{i=1}^{N}(\rho(||\Psi(X_i, Q, x_i)||^2))\} \tag{4.2}$$

In the experiment section, we keep all the frames with $w \geq 0.35$ for the subsequent recognition tasks.

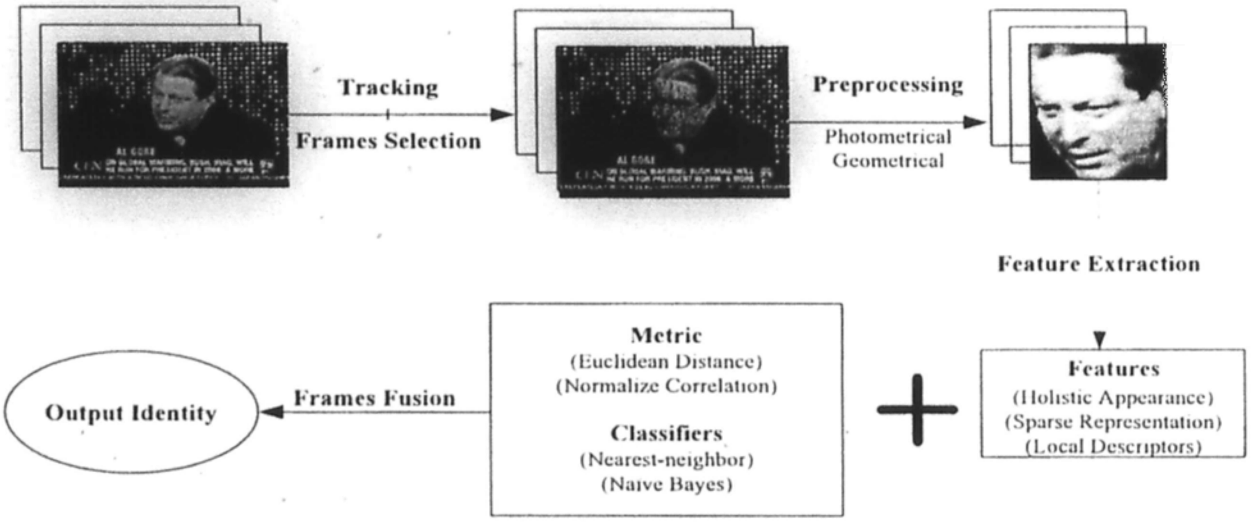


Figure 4.2: Flow diagram of the video recognition

4.3 Recognition From Image Sets

In this section, we consider face recognition with sample images from k individuals that have been properly cropped and normalized from videos. Given the face tracking outputthe procedure of video based recognition is illustrated in Figure. 4.2, which consists of three stages: firstly, a feature extraction component is used to extract feature descriptors for each frame; secondly, a frame based recognition process is performed, the recognition result for each frame is given with the usage of a measurement function and a classifier; last but not least, a frame fusion strategy is utilized, which integrates all the individual frame recognition for the final identity decision.

In the following subsections, we first introduce the feature descriptors we used, which include holistic intensity image, local patch descriptors and sparse representation. Then, the classifiers and the related fusion strategies for recognition are also described.

4.3.1 Holistic Descriptors

As for the holistic feature based methods, each face image is represented as a single high dimensional vector by concatenating the gray values of all pixels in the face. The advantages of this representation are two folds. First, it implicitly preserves all the detailed texture and shape information that are useful for distinguishing faces. Second, it can capture more global aspects of faces than local feature-based descriptions.

Starting from the successful low dimensional reconstruction of faces using PCA projections [84], eigenface has been one of the most widely used representation for the vision tasks, for examples, recognition, tracking, e.t.c. The intrinsic thought behind such a representation is that the highly structured global face appearance is assumed to reside on a subspace of much lower dimension. However, the main target of PCA is for dimensional reduction and compression, thus it may not lead to optimal discrimination. To extract the discriminant features, LDA [72] is further applied to the top-level principal components. It pursues a linear transform W that maximizes the ratio of between-class scattering and within-class scattering.

$$W = arg \max_{W} \frac{W^T S_b W}{W^T S_w W}. \tag{4.3}$$

Suppose the training set has n images from C different classes, denoted by $\{(I_1, c_1), (I_2, c_2), \ldots, (I_n, c_n)\}$, where c_i is the label of the i-th sample. Denote the principal component projection for I_i by x_i, then the between-class scatter matrix S_b and the within-class scatter matrix S_w are respectively defined as:

$$\begin{aligned} S_b &= \tfrac{1}{n} \sum_{k=1}^{C} n_k (m_k - m)(m_k - m)^T, \\ S_w &= \tfrac{1}{n} \sum_{k=1} \sum_{i:c_i=k} (x_i - m_k)(x_i - m_k)^T, \end{aligned} \tag{4.4}$$

where m_k is the mean vector of the k-th class, and m is the total mean. The optimal projection matrix W can be obtained by solving a generalized eigenvalue problem:

$$S_b W = S_w W \Lambda_w. \tag{4.5}$$

It is well known that LDA tend to suffer from the singularity of S_w in a high dimensional space. A series of improved LDA algorithms are proposed to address such difficulty, including PCA+LDA [113], Enhanced Fisher Model[58], Unified Subspace Analysis[91], and Null-space LDA[17]. They resolve the problem at the expense of losing information in either principal subspace or null space of S_w. Meanwhile, dual-space LDA [92] and Regularized LDA [57] tackle the singularity problem while retain the principal subspace and its complement. In the following part of the thesis, we take the Regularized LDA as the holistic appearance descriptor.

With a learned Regularized LDA model W, we can compute the discriminant feature vectors by

$$y = W^T x. \tag{4.6}$$

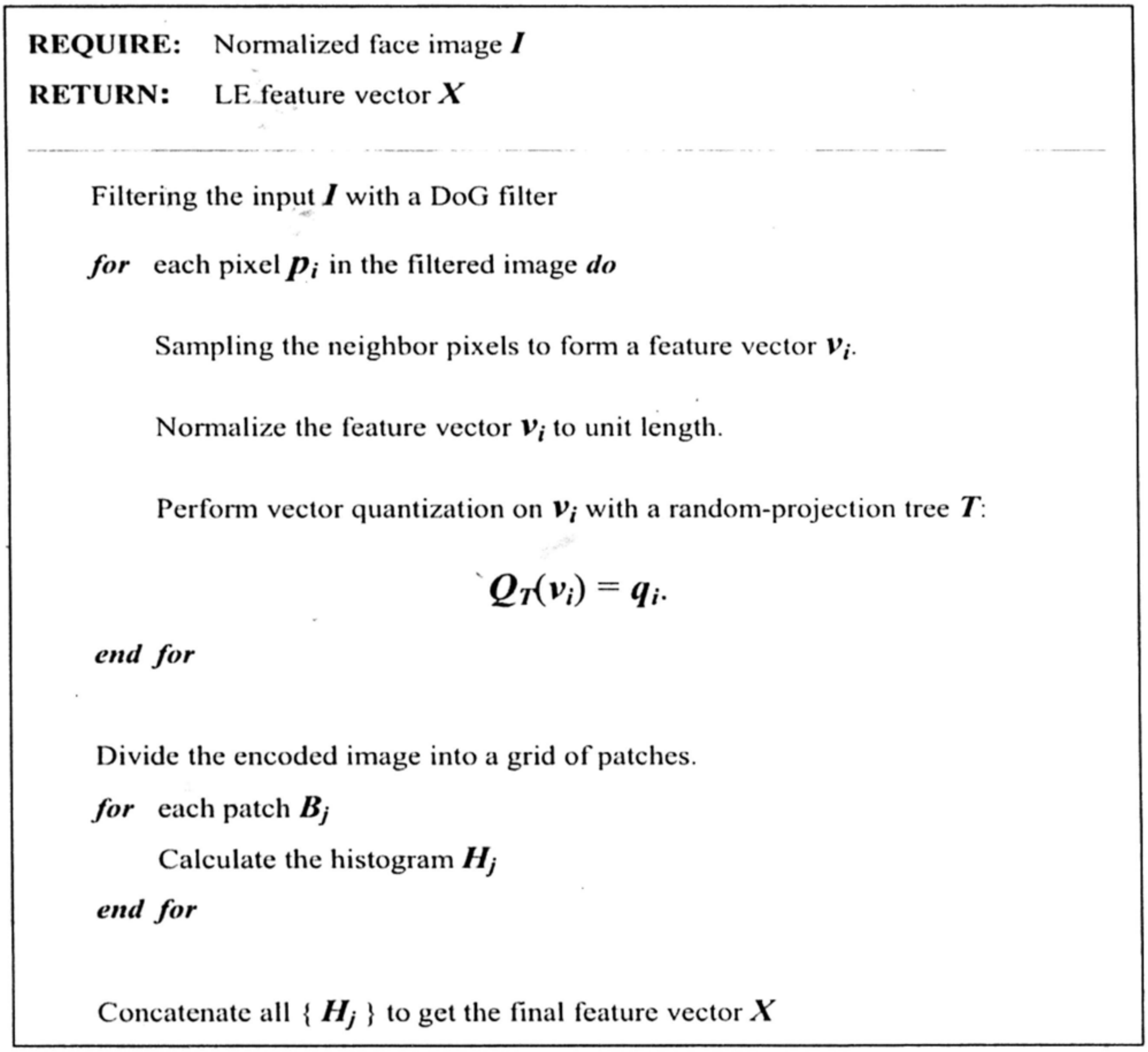
Hence, for two faces denoted by x_1 and x_2, the distance measurement is given by $d = dist(W^T x_1, W^T x_2)$. The different implementations of the distance functions $dist(\cdot, \cdot)$ can influence the recognition performance dramatically, as demonstrated in [68]. In this chapter, we take the following two types of distance measurement: Euclidean distance and normalize correlation. At the final step, a Nearest Neighbor (NN) classifier is used to assign a test sample with the class label of the closest training sample.

4.3.2 Local Descriptor

Local methods that use local facial features for face recognition is a relatively mature approach in the field with a long history [35],[70],[14]. Compared with holistic methods, local descriptors may be more suitable for the task of recognition due to the following observations: Firstly, in local descriptors, the original face is represented by a set of low dimensional local feature vectors, rather than one single full high dimensional vector, thus the curse of dimensionality can be alleviated from the beginning. Secondly, local methods provide additional flexibility to recognize a face based on its parts, thus the common and class-specific features can be easily identified. Thirdly, different facial features can increase the diversity of the classifiers, which is helpful for face recognition.

Despite those advantages, the local methods need to enforce a global configuration to perform the recognition from a macroscopic perspective. The incorporation of global configurational information in faces is extremely critical for the performance of a local method. Generally, there are two ways to pursue the combination. First, the global information can be explicitly embedded into the algorithm using such data structure as graph, where each node represents a local feature, while the edge connecting two nodes accounts for the spatial relationship between them. Face recognition are then formulated as a problem of graph matching. Alternative approach to incorporating global information is to concatenate the local descriptors into a long vector. The metric and classifiers used in the holistic approaches are used to achieve the final decision.

Currently, the state-of-the-art recognition performance for still images is achieved with the usage of local descriptors [43][78] for face representation. Local Binary Pattern (LBP) [1] is one of the most representative approaches. LBP encodes the relative intensity magnitude between each pixel and its local neighbor pixels. Thereafter, the micro-structure of a face is described by the histogram of LBP from the correspondent facial patch. Such an encoding scheme and feature descriptor is less sensitive to the monotonic photometric change. Meanwhile, it share the merit of high efficiency computation. However, the LBP-based approaches [79][96][107] suffer from the uneven distribution since the handcrafted original. Meanwhile, they need a tradeoff on the size of codebook between the discrimination and robustness in a high dimension space.



```
REQUIRE:  Normalized face image $I$
RETURN:   LE feature vector $X$

Filtering the input $I$ with a DoG filter
for each pixel $p_i$ in the filtered image do
    Sampling the neighbor pixels to form a feature vector $v_i$.
    Normalize the feature vector $v_i$ to unit length.
    Perform vector quantization on $v_i$ with a random-projection tree $T$:
        $Q_T(v_i) = q_i$.
end for

Divide the encoded image into a grid of patches.
for each patch $B_j$
    Calculate the histogram $H_j$
end for

Concatenate all { $H_j$ } to get the final feature vector $X$
```

Figure 4.3: Learning Based Descriptor Extraction

In this chapter, we utilize a learning-based encoding scheme for the local descriptor construction, which is originally proposed by Cao *et. al.* [14]. The core idea of the proposed approach is to encode the local micro-structures of the face into a set of discrete codes. Thus the learned codes are more uniformly distributed and the resulting code histogram can achieve much better discriminative power and robustness tradeoff than existing manually tuned encoding approaches. To keep the integrity of thesis, we give a brief introduction of the learning-based(LE) descriptor.

Figure. 4.3 shows the he algorithm for LE descriptor extraction from an input image. The input image is fed into a DoG filter (with $\delta_1 = 2.0$ and $\delta_2 = 4.0$) to alleviate the illumination variations. At each pixel, we sample its neighbor pixels in the ring-based pattern to form a low dimension feature vector (r-neighbor pixels on the ring of radius r). In the following experiment configuration, we take the radius r to 2. Therefore, together with the value of the central pixel, a 25-dim feature vector is built up for each pixel. After the local feature construction, we normalize the feature vector into unit length. Such normalization combined with DoG filtering makes the local feature vectors invariant to local photometric affine change.

At the quantization step, the normalized feature vector is encoded into discrete codes. Different from the handcrafted coding schemes, the LE encoder is trained for face images with an unsupervised approach: random-projection tree [32]. At the training stage, a random-projection tree is built up based on the uniform criterion, which means that the samples distribution on each leaf node is as even as possible. In our experiment, we train a 256 nodes random-projection tree from annotated face images from LFW [44]. For each of the 25-dim feature vector, it is quantized to the discrete code that range from 1 to 256.

After the encoding step, the input image is transferred into a code image. Following the method described in LBP [1], the encoded image is divided into a grid of patches. In our experiment, we take a patch grid of 6 × 6 for the normalized input image with resolution 80 × 90. A histogram (256 bins) of the LE codes is built up for each patch and the patch histograms are concatenated to form the descriptor for the whole face image. The total length of the final descriptor is 9216.

Generally, it is hard to directly use such a high dimensional feature vector for the recognition task. A high dimensional not only limits the number of faces to be processed, but also increase the computation load for face recognition. Thus, we apply PCA to compress the concatenated histogram and use the compressed descriptor as the final LE descriptor. In all the following experiments, we use PCA for dimension reduction with 98% energy preserved. The measurement functions and classifiers used in the local descriptors based approach are the same as those for the holistic approach.

4.3.3 Sparse Feature and Recognition

In the signal processing community, the problem of pursuing sparse linear representations with respect to an overcomplete dictionary of base signals has seen a recent surge of interest [25][112]. The core problem of these approaches is to find an optimal representation that is sufficient sparse. It can be sufficiently computed by convex optimization, which is similar to the Lasso in statistics [80][25], penalizes l^1-norm of the coefficients in the linear combination.

In [97], Yi Ma *et.al.* take the advantage of sparse representation for classification in the context of automatic face recognition. The underlying rationality to use sparse representation for classification is that: *Suppose we have sufficient training samples for each class, it will be possible to represent the test samples as a linear combination of just those training samples from the same class. This representation is naturally sparse, involving only a small portion of the overall training images.* With comparison to the scenario of recognition for still images, the video based face recognition provides more training samples, where the sparse assumption is more suitable.

Given sufficient training samples of the i-th object class, $C_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, a new test sample $y \in \mathbb{R}^m$ from the same class will be approximately lie in the linear span of the training samples of C_i.

$$y = \alpha_{i,1}x_{i,1} + \alpha_{i,2}x_{i,2} + \ldots + \alpha_{i,n_i}x_{i,n_i}, \tag{4.7}$$

with the linear combination coefficient $\alpha \in \mathbb{R}$.

For each y, it is represented by an overcomplete dictionary that contains the entire training set of n samples from k classes:

$$C = [C_1, C_2, \ldots, C_k] = [x_{1,1}, x_{1,2}, \ldots, x_{k,n_k}]. \tag{4.8}$$

Therefore, the linear representation of y can be rewritten in terms of C as:

$$y = C\alpha \in \mathbb{R}^{\hat{m}}. \tag{4.9}$$

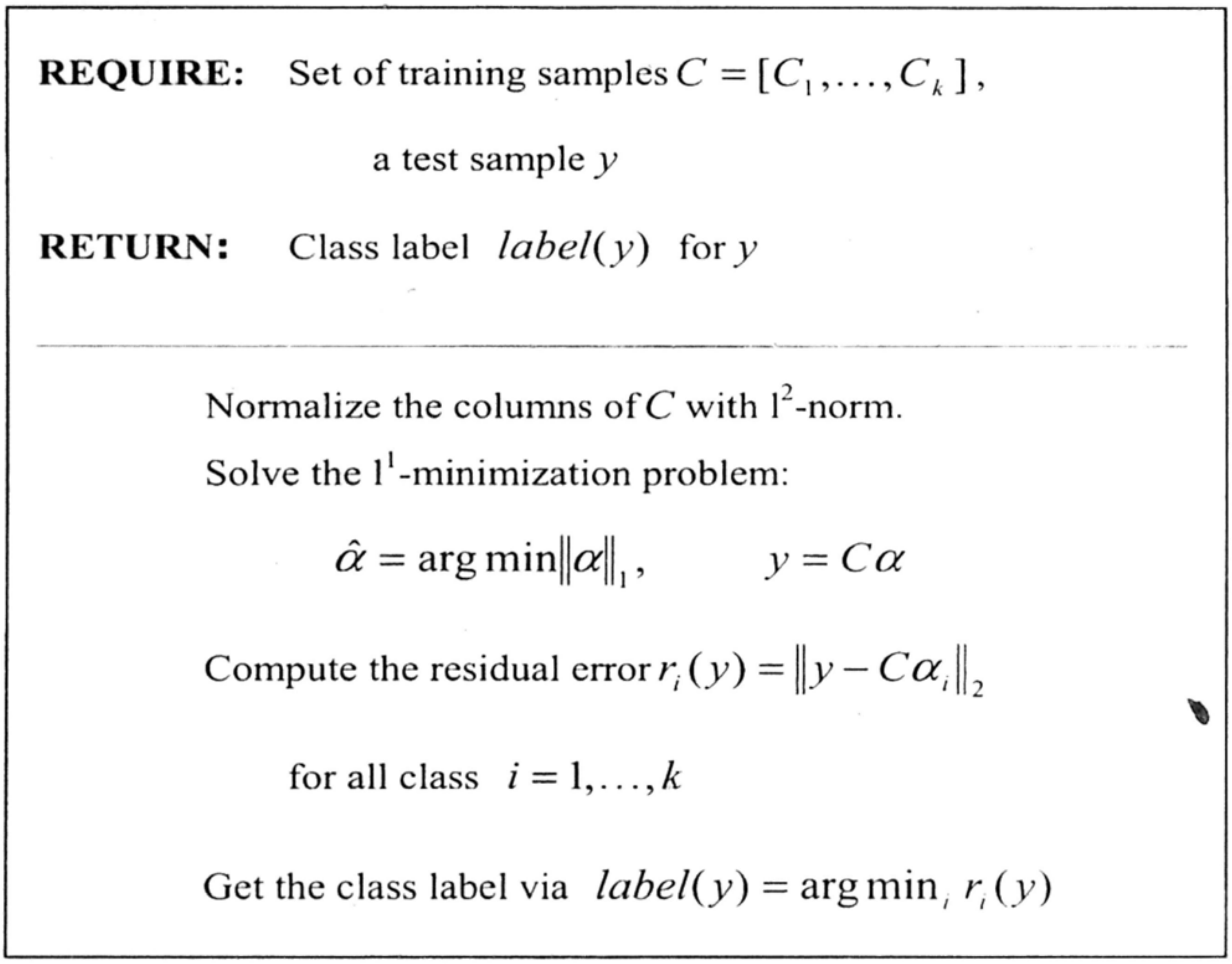
Seeking a sparse solution α to Eq. (4.9) is done by solving the following l^1-minimization problem:

$$\hat{\alpha} = \arg\min \|\alpha\|_1, \quad y = C\alpha. \tag{4.10}$$

This problem can be solved in polynomial time by the standard linear programming method[18].

After get the sparse representation $\hat{\alpha}$ for a test sample y via Eq. (4.10). In an ideal case, the nonzero entry in $\hat{\alpha}$ will be correspondent to the column of C from a single class C_i. However, errors arising from modeling and noise may lead to small nonzero entries associated with multiple classes.

For each class i, denote $\alpha_i = [0, \ldots, \alpha_{i,1}, \ldots, \alpha_{i,n_i}, 0, \ldots]$ that selects the coefficients associated with the i-th class for a test sample y. Using α_i,we can approximate y as $\bar{y}_i = C\alpha_i$. Therefore, we classify y based on these approximations by assigning it to the class that minimizes the residual between y and $\bar{y}_i$:



```
REQUIRE:  Set of training samples $C = [C_1, \dots, C_k]$,
              a test sample $y$
RETURN:   Class label $label(y)$ for $y$
------------------------------------------------------------
    Normalize the columns of $C$ with $l^2$-norm.
    Solve the $l^1$-minimization problem:
        $\hat{\alpha} = \arg\min \|\alpha\|_1, \qquad y = C\alpha$
    Compute the residual error $r_i(y) = \|y - C\alpha_i\|_2$
        for all class $i = 1, \dots, k$
    Get the class label via $label(y) = \arg\min_i r_i(y)$
```

Figure 4.4: Recognition Based on Sparse Representation

$$\min_i r_i(y) = \|y - \hat{y_i}\|_2. \tag{4.11}$$

The overall recognition process for one single frame is summarized in Figure. 4.4.

4.3.4 View Based Models for Pose Accommodation

A view based classification strategy is taken into the training process, to handle the huge appearance variance owing to the out-of-plane motion. For each of the cropped face samples, no matter from training set or query set, a view label is assigned based on its 3D motion parameters. In our experiments, five views are exploited: $\{-70°, -45°\}$, $\{-45°, -20°\}$, $\{-20°, 20°\}$, $\{20°, 45°\}$, and $\{45°, 75°\}$. With the view based strategy, all the training samples have the same view label are trained and the classification for each query frame is also performed on the trained models with the same view label.

4.3.5 Decisions Combination

Given the classification results from each query frame, a decision fusion scheme is needed to get the final decision for the overall video. In the following experiments, two combination strategies are used. The first one is a voting scheme, where the single classification decision from each frame are collected to build a class histogram. The class with maximum votes is assigned to the whole image sequence. Beside the rank-1 classification results, we also offer the rank-2 (class with second most votes) and rank-3 (class with third most votes) classification results.

Another combination strategy is defined on the measurement function. Given the training samples from m object classes, $P = [P_1, \ldots, P_m]$. P_i contains all the videos for i-th class. For an input query video with n frames,$Q = [q_1, \ldots, q_n]$, we define a sample-class distance function $\mathbf{d}_j^k$ for the i-th query frame and k-th class as:

$$\mathbf{d}_j^k = min\{dist(q_k, p_l) | p_l \in P_k\}, \tag{4.12}$$

where $dist(\cdot)$ is one of the two measurement functions: Euclidean distance and normalized correlation. At the next step, we get a measurement function defined between the overall query frames and a specific class k:

$$\mathbf{d}^k = \frac{1}{n_k} \mathbf{d}_j^k. \tag{4.13}$$

In the above equation, n_k is the number of query frames that can retrieve the sample-class distance function. In the design of , we prefer the average distance over samples to the sum distance over samples. This preference is based the following observation: With the exploitation of view based models, the query frame number n_k for each class k might be different. It is possible that some view labels are not contained in the training samples of the k-th class. At a final step, we assign the query video with class label $\hat{k} = \arg\min_k {}^k$.

However, we only exploit the voting scheme for decisions combination for the recognition approach based on sparse representation. Since sparse representation attempts to solve the representation coefficients together with all the training samples. It is meaningless to store the distance to all the object classes, since the sparse coefficients only concentrate on few samples from even less classes. For most of the object classes, the correspondent coefficients keep zero entry.

4.4 Experiments

In this section, we present experiments on publicly available databases for video based face recognition, which demonstrate the efficiency of the proposed recognition framework. With the accurate face localization together with pose extraction, we can achieve the state-of-the-art recognition performance even with the using of simple holistic appearance features.

We will first introduce the tracking configuration of the proposed framework and the related face cropping process, which act as a preprocessing step. Then we test the recognition performance of the proposed approaches on specifically designed benchmark datasets. The role of feature extraction in the proposed recognition framework is examined with a series of comparative experiments, together with the performance analysis. The overall performance on the real-world videos outperform the previous best reported approach on the same data set [48], even with a simple holistic feature.

4.4.1 Video Datasets

The video set used in our experiment contains quantities of noisy real-world videos crawled from YouTube, which is public available [48]. The dataset consists of 46 identities with 1910 video segments. The typical resolutions of video varies from 240 × 180 to 320 × 240. Since our approach needs a frontal face frame for automatic initialization, we omits the video clips failed to initialize the tracking approach. The remaining 1239 video clips are tracked and used for the recognition evaluation. This video set is challenging for both tracking and recognition processes since videos exhibit large variations in face pose illumination, expression, e.t.c.

4.4.2 Track and Crop Faces

The proposed non-rigid 3D tracking is automatically initialized in an image with frontal face: first the face rectangle is automatically detected by a face detector and then 2D facial features are located by the face alignment algorithm [115, 105]; given the localized 2D features and their corresponding 3D vertex index on the deformable 3D shape model, the shape and pose parameters are estimated using the method in [42]. What should be mentioned here is that our 3D tracker's parameters are fixed for all the videos used in the experiments, no matter whether they come from indoor scenes or are crawled from Internet.

To crop a normalized face, we locate the eyes centers and the mouth center for each well tracked frame. Geometric normalization and histogram equalization are performed sequentially and the normalized image size is rescaled to 80×90, which is a common size for image based recognition. Figure. 4.5 shows the examples of the well cropped faces.

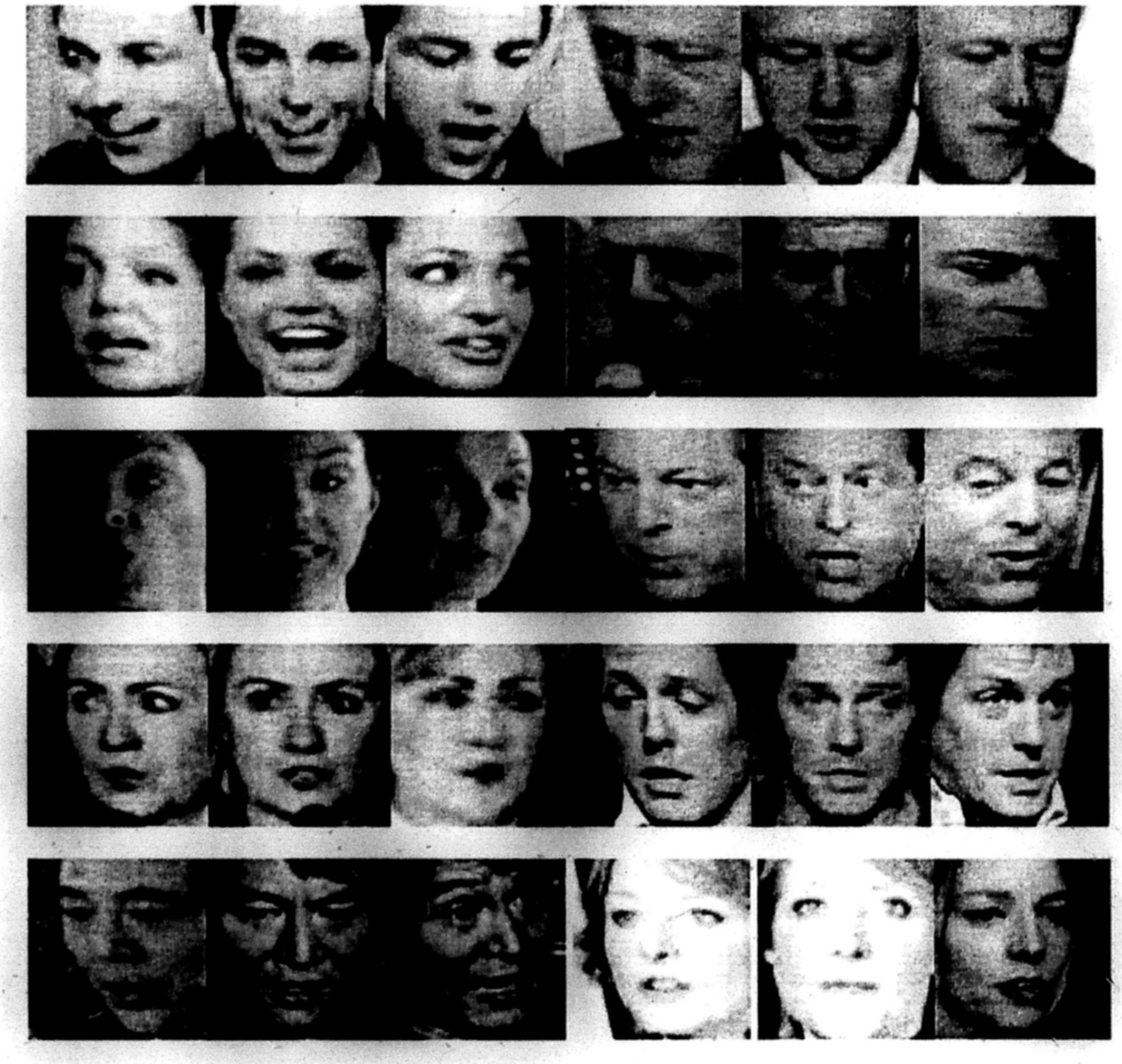


Figure 4.5: Cropped samples from the Youbube Dataset

Feature Descriptor	Voting	Average Distance
Regularized LDA + l^2-*norm*	81.52%	59.78%
Regularized LDA + *normalized correlation*	83.70%	80.43%
LE + l^2-*norm*	83.70%	66.30%
LE + *normalized correlation*	88.04%	85.87%
sparse representation + l^2-*norm*	84.78%	N/A
sparse representation + LE +l^2-*norm*	80.43%	N/A

Table 4.1: Accurate Recognition Rate (ARR) with Different Feature Descriptors

4.4.3 Recognition on Video Datasets

In the following recognition experiments, the accurate recognition rate (ARR) is used as the performance indicator of the recognition approach. We first verify the role that feature descriptors play in our recognition framework.

Three types of features are tested on the same video dataset with other experiment configurations keep unchanged: Regularized LDA, LE, and sparse features. For the data preparation, we randomly choose 4 videos for each of the 46 identities from the Youtube dataset, 2 videos for training and 2 for test. Two standard measurement functions are used for the first two features: normalized correlation and l^2-norm. Meanwhile, two types of different decision fusions are exploited. For each train/test video, we take only the first 100 well tracked frames to represent the overall video.

As shown in Table. 4.1, the combined feature of sparse representation and LE descriptor is achieved by replacing the original croppped sample with the extracted LE descriptors. Using LE descriptor with normalized correlation measurement achieve the best recognition result. However, with the precise facial feature extraction, the holistic feature perform as well as we expected. It has also been noticed that the performance of combining LE descriptor and sparse representation is lower than only using sparse representation. This conclusion coincides with the assertion that LE descriptor seeks to encodes the local information as even as possible, the l^1-*norm* minimization on LE is less

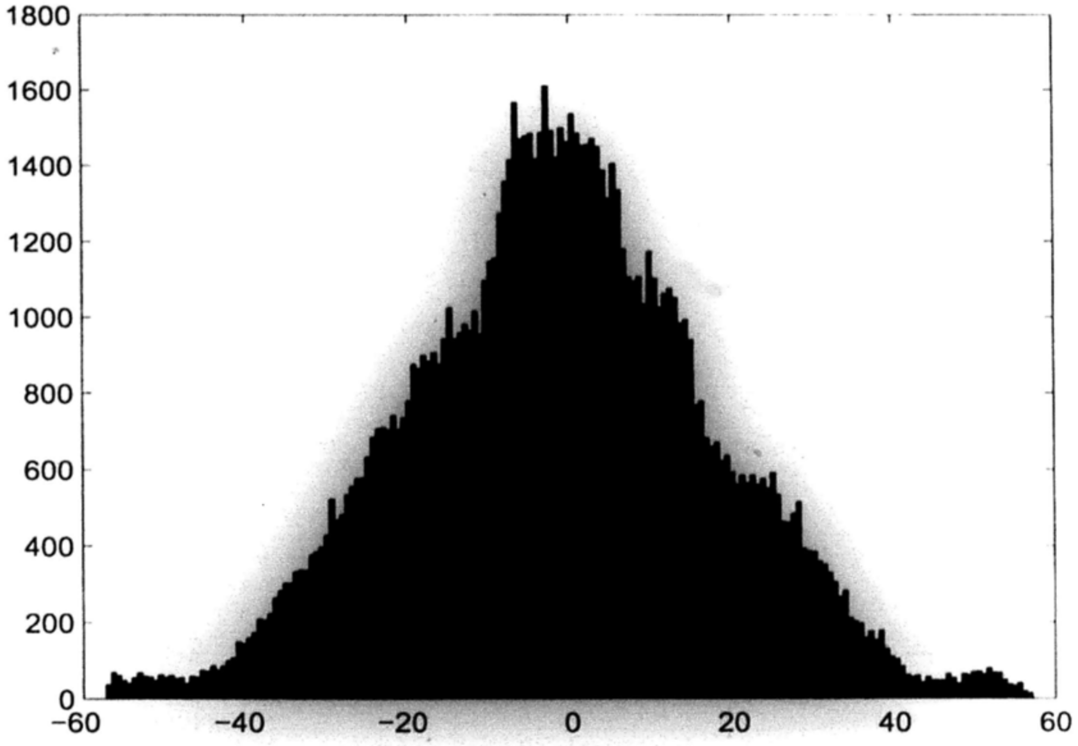


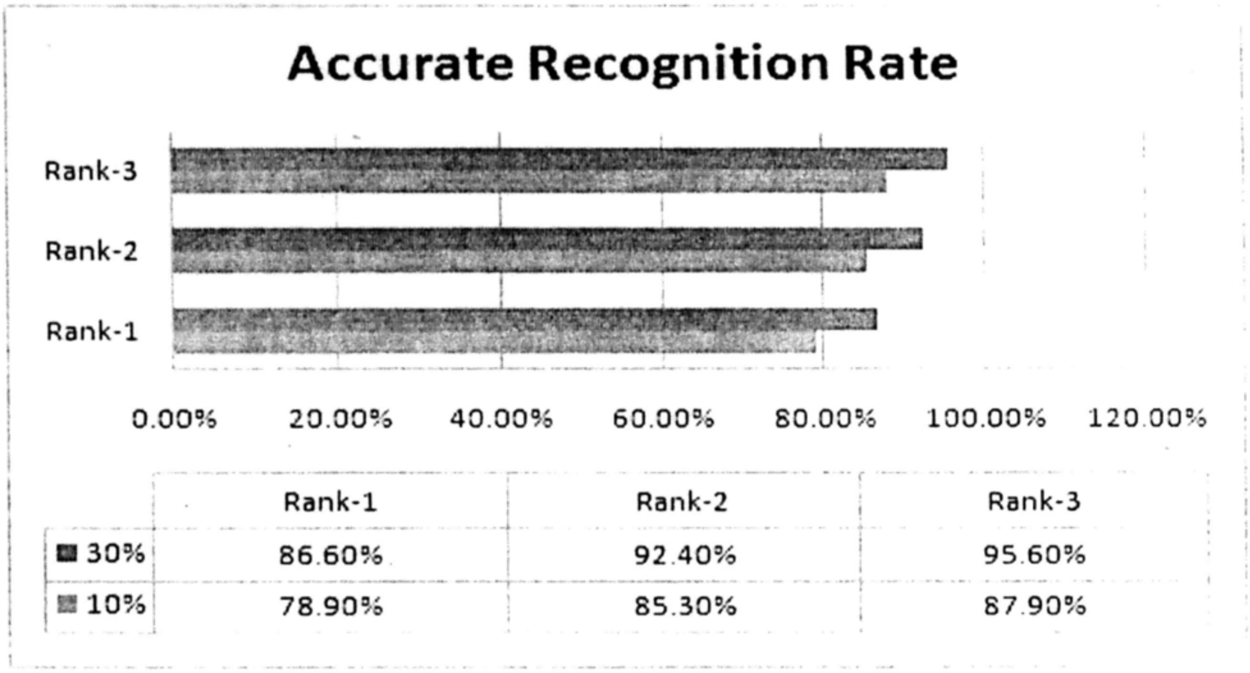
Figure 4.6: The pose distribution among training samples

discriminative than the original samples.

The recognition results on the Youtube dataset are shown Table. 4.2, in which the comparison results in the first two columns are cited from [48] on the same video dataset. In our experiments the training and testing sets are randomly partitioned in a similar way to [48]: ARR of 78.9% is achieved with a random partition with 10% videos used for training (132 clips). The training and testing videos include all the 46 persons. To the best of our knowledge,

	2D Tracking Voting	2D Tracking HMM	3D Tracking Voting
Persons	35	35	46
Videos	1200	1200	1239
ARR	62.42%	71.24%	78.9%

Table 4.2: Comparison of ARR on Youtube Dataset



	Rank-1	Rank-2	Rank-3
■ 30%	86.60%	92.40%	95.60%
▨ 10%	78.90%	85.30%	87.90%

Figure 4.7: ARR on with Different Train/Test Partition

ARR of 71.24% in [48] is the best recognition performance on the Youtube data, where around 1200 video clips with 35 identities are used. However, since the authors in [48] could not offer their detailed training and testing partition. So such an comparison is less meaningful. Moreover, when more persons are included in the recognition experiment (46 persons with respect to 35 persons in [48]), while the total number of videos keeps unchanged, the problem becomes more harder.

Obviously, the ratio of the training and testing is a key parameter for the final recognition output. To demonstrate the influence of the amount of training samples on the final recognition result, we give the recognition results in Table. 4.7 when 10% and 30% of the dataset are partitioned as training set:

The figures in the above table demonstrate that when more clips are used in the training data set, the ARR rate soars up from 78.9% to 86.6%, while the misclassified video clips reduces from 232 to 91. When rank-2 and rank-3

	Frontal Only	3 Views	5 Views
Regularized LDA	84.0%	86.6%	85.9%
LE + Regularized LDA	89.2%	90.27%	89.9%

Table 4.3: ARR with Different Views

classification is taken into consideration, we can achieve a high ARR close to the correct decision. When the misclassified videos are retrieved, we find that the major cause of the misclassification is that the usage of the confidence measure cannot always select the well tracked frames when blur effect, agile motion and severe illumination variation exist in the video clip. As a result, the tracking is drifted away and the voting scheme cannot guarantee to select the correct identity.

The pose distribution among the YouTube dataset is shown in 4.6, most of the samples are concentrated in the range $\{-20^\circ, 20^\circ\}$. To investigate the robustness of the recognition results and the validation of the view-based models, we test the recognition performance with different view models: only first view model; three view models including $\{-45^\circ, -20^\circ\}$, $\{-20^\circ, 20^\circ\}$, $\{20^\circ, 45^\circ\}$; all five view modes. We randomly select 30% of the videos (363) as training set, while the remainder 70% videos (853) are used for test.

In Table.4.3, we give the recognition results with Regularized LDA and LE. With an accurate face extraction approach together with a precise view partition, the recognition results with usage of a simple holistic appearance feature are comparable to the recognition results using a state-of-the-art feature descriptor.

We are interest in the variation of ARR when we increase the used view-based models. For both of the two features, the best ARR is achieved when 3 views are used for recognition. When 5 views are used, the ARR declines slightly. A reasonable interpretation for the performance degeneration is as following. For some specific persons in the test set, they do not have training samples in the side views models, due to the random partition scheme. Thus, the view based model labels the sample with a wrong identity for that frame, which leads to a wrong voting result at the decision combination stage.

4.5 Conclusion

In this chapter, we propose a video face recognition approach based on 3D non-rigid face tracking. The proposed 3D tracker can provide both accurately cropped face and reliable face view information, and thus significantly improves the performance of face recognition. The confidence provided by the tracker also serves as a criterion for robust face recognition in video sequence. With the well cropped samples, we can achieve the state-of-the-art recognition performance even with the usage of simply holistic features. Extensive experiments have been carried out on the videos crawled from Internet, which demonstrate the superior performance of our approach over existing approaches.

Chapter 5

Performance Driven Face Animation

In this chapter, a performance driven 3D face animation system is proposed. The proposed approach consists of two key components: a robust non-rigid 3D tracking module and a MPEG4 compliant facial animation module. Firstly, the facial motion is tracked from source videos which contain both the rigid 3D head motion (6 DOF) and the non-rigid expression variation. Afterward, the tracked facial motion is parameterized via estimating a set of MPEG4 facial animation parameters(FAP). As the final step, these FAP values are transferred to the MPEG4-compliant face model for the animation purpose. Compared with the recent works on performance driven animation systems, the potential advantages are two folds: Firstly, the non-rigid face tracking provides a global motion for the animation, which is more robust with comparison to the approaches built on 2D tracking of few control points; Secondly, the proposed tracking and animation system has a strong generalization ability and can be used in the indoor environment with no additional assumptions.

5.1 Introduction

Facial animation is highly demanded in the applications of 3D games, interactive human/computer softwares and movies, for the human face can express a wide gamut of emotions and expressions that can vary widely both in intensity and meaning [46][71]. Traditionally, deformation of a face model is designed by experienced animator with tedious hand sculpture. Constructing and simulating such a model has been proven to be a difficult task because of the subtlety of face skin motions. Contrarily, performance driven facial animation [93] has been a hot topic since it allows actors to express content and mood naturally for target animation controlling, and the resulting animations have a degree of realism that is hard to obtain from the hand crafted works.

Since the pioneering work of [71], quantities of efforts were provided to increase the realism for ensuring the appealing interaction with humans. In [93], an approach for building realistic human head model based on photographic texture mapping has been suggested. Based on this result, a range of systems has been developed that analyze the expressions of a human performer and transfer the correspondent facial animation onto the face models. In [37][75], a morphable model is built by fitting a generic face model to multiple photographs of one facial expression using manually labelled feature points; the model is afterwards used to track video sequences of the same person. In [26], an animation system is designed for teleconferencing based on landmark tracking in videos. This process is aided by a deformable 3D mesh model that has been obtained from a 3D laser scan.

For the success of a performance driven animation system, it needs to tackle problems arising from the following aspects: What kind of approaches should be used to obtain the model of facial motion? How to drive a 3D facial motion animation using tracked facial landmarks.

For tracking landmarks of face, adding markers on human face was used

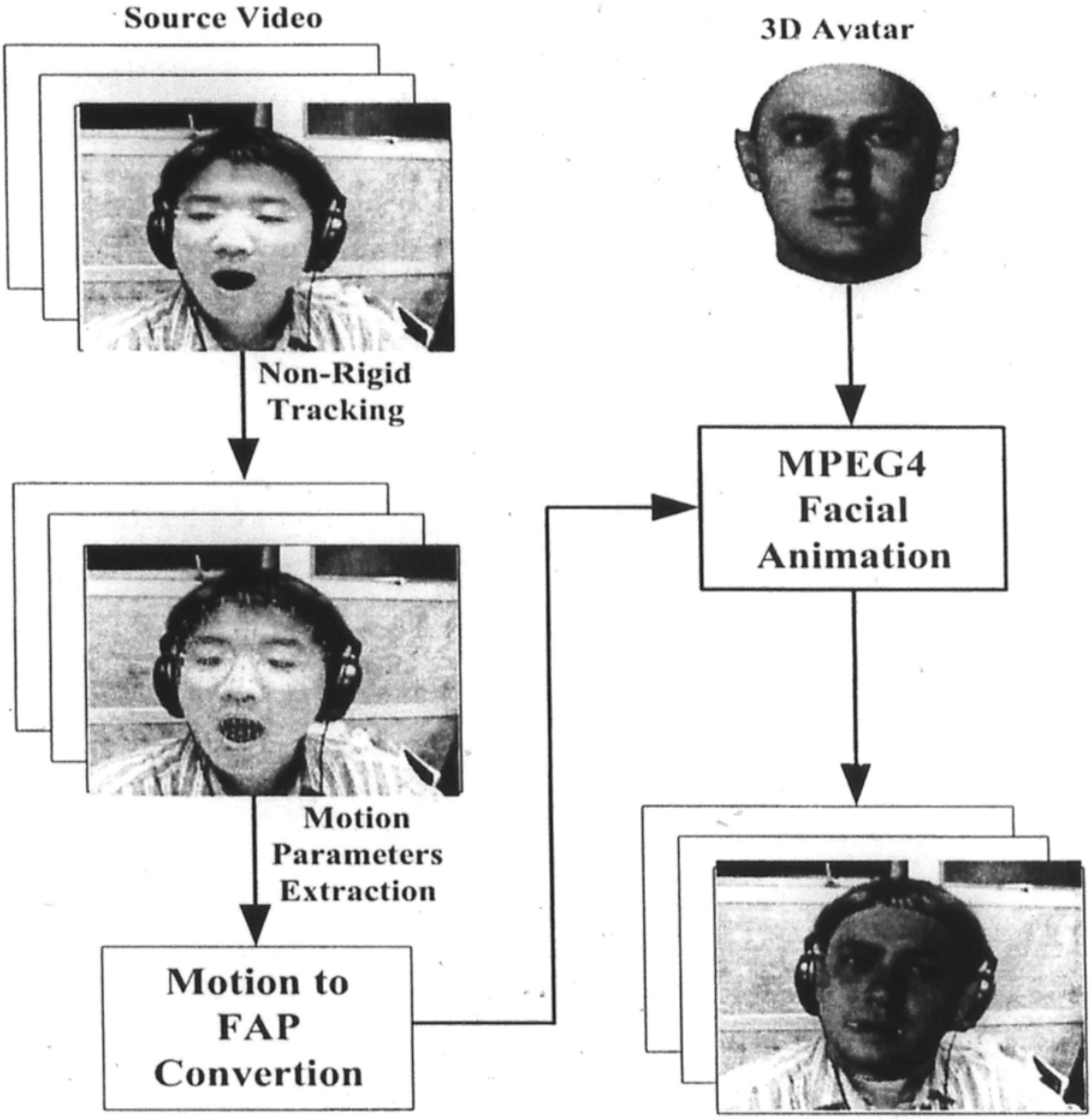


Figure 5.1: Flow diagram of the animation algorithm

in many of early work, for example, using a set of colored dots [37] [94]. Once the position of marker determined, the facial motion features can be easily derived from image or video sequence. However, this technique is limited only suitable for the case in there markers on face existed or allowed to be used. The approaches in a later stage mainly concern face tracking in unmarked face of video. However, the traditional approach[15] rely on the 2D feature tracking to extract the source control, which limits the application scope only to the near frontal face.

To handle it to map the facial motion of the performer onto meaningful animation of the target model. There have been several different approaches to map the facial motion from the source to the target model. Generally they can be divided into two categories: the usage of a blend shape system [46][15] for target animation synthesis or directly motion retargeting on the target motion model. Expression cloning [69] is a typical technique in the second category, it directly maps an expression of the source model onto the surface of the target model, while no assumptions are made on the source/target models about the mesh geometry or connectivity. Given the features correspondence between source and target models, the motion transferring from the source models to the target models is carried with the criterion that the motion vector should be properly adjusted to account for the local shape of the models.

Inspired by Chuang's work [19] to mapping a performance directly to a target blend shape set, we present a novel 3D facial animation framework where deformation of the target model is controlled animators in front of the web-camera: the non-rigid 3D facial motion is captured by registering a 3D morphable model to each input frame; afterward, the rigid pose and non-rigid expression parameters is extracted and transferred to the target animation model, in which the shape blending technique is used to synthesis the final result.

Compared with the recent vision based animation systems in [19][15][46], the usage of 3D deformable tracking for animation driven has the following advantages: Firstly, the global facial motion is tracked instead of few control points for the animation, which is more robust and not constrained to the near frontal pose; Secondly, the intrinsic decoupling of the pose and expression parameters makes us only need to focus on the non-rigid facial motion, which guarantee the animation procedure to be more accurate; Thirdly, the usage of a 3D motion data for FAP calculation reduces the ambiguity arising from the 2D tracked based approaches. Meanwhile, the MPEG-4 compatible motion retargetting scheme has a strong generalization ability. No adaption is required for different control sources. Experiments show that our system produces more vivid face motion animation in a large pose range.

5.2 Animation System Description

Given a video sequence as input, the proposed system is used to drive the animation of a 3D model. Figure. 5.1 illustrates the flow diagram of whole animation algorithm. Firstly, non-rigid motion of the source video is captured by a non-rigid 3D face tracker. Then, the extracted motion vectors are used to calculated expression parameters which are based on the definition of MPEG-4 FAP. Finally, expression parameters are transferred to the target model for animation purpose. The performance of non-rigid tracking and mesh animation is real-time. Meanwhile, the only requirement for the animation system is a neutral face input at the first frame.

5.2.1 Non-rigid 3D Tracking

Given a collection of image observations $\{\mathbb{I}\}$, non-rigid 3D face tracking at frame t is formulated as the Maximum A Posterior (MAP) estimation in Bayesian framework where the tracking state consists of non-rigid 3D shape

X_t and rigid pose Q_t:

$$\begin{aligned}\{\hat{X}_t, \hat{Q}_t\} &= \arg\max_{X_t, Q_t} p(X_t, Q_t|\{\mathbb{I}\}) \\ &\propto p(\{\mathbb{I}\}|X_t, Q_t) \cdot p(X_t).\end{aligned} \tag{5.1}$$

$p(X_t)$ denotes the prior constraints of 3D face deformation on X_t, $p(\{\mathbb{I}\}|X_t, Q_t)$ is the likelihood distribution which describes the conditional probability of image observations $\{\mathbb{I}\}$ given the tracking state $\{X_t, Q_t\}$. The tracking algorithm consists of three key components:

1. *the prior shape model* $p(X)$, *it characterizes the 3D shape variation from a set of training samples;*

2. *the likelihood model* $p(\{\mathbb{I}\}|X, Q)$, *which is modeled based on the 2D-3D features correspondences. The features set consists of the off-line trained semantic features, face silhouette features and online tracked low-level features;*

3. *the robust estimation algorithm which obtains reliable results in the case of significantly noisy image observations. We make use of an hierarchical optimization strategy together with robust estimation techniques to fulfill the task.*

The generalization capability is guaranteed with the usage of deformable shape model together with the off-line trained distinctive facial features. Figure. 5.2 shows the results of our tracking approaches for one performer. Detailed algorithm can be referred in Chapter. 3.

5.3 Controlling of Target Model

Given the source motion extracted from videos, the key problem for target model animation is to retargeting the motion where the topology of the target

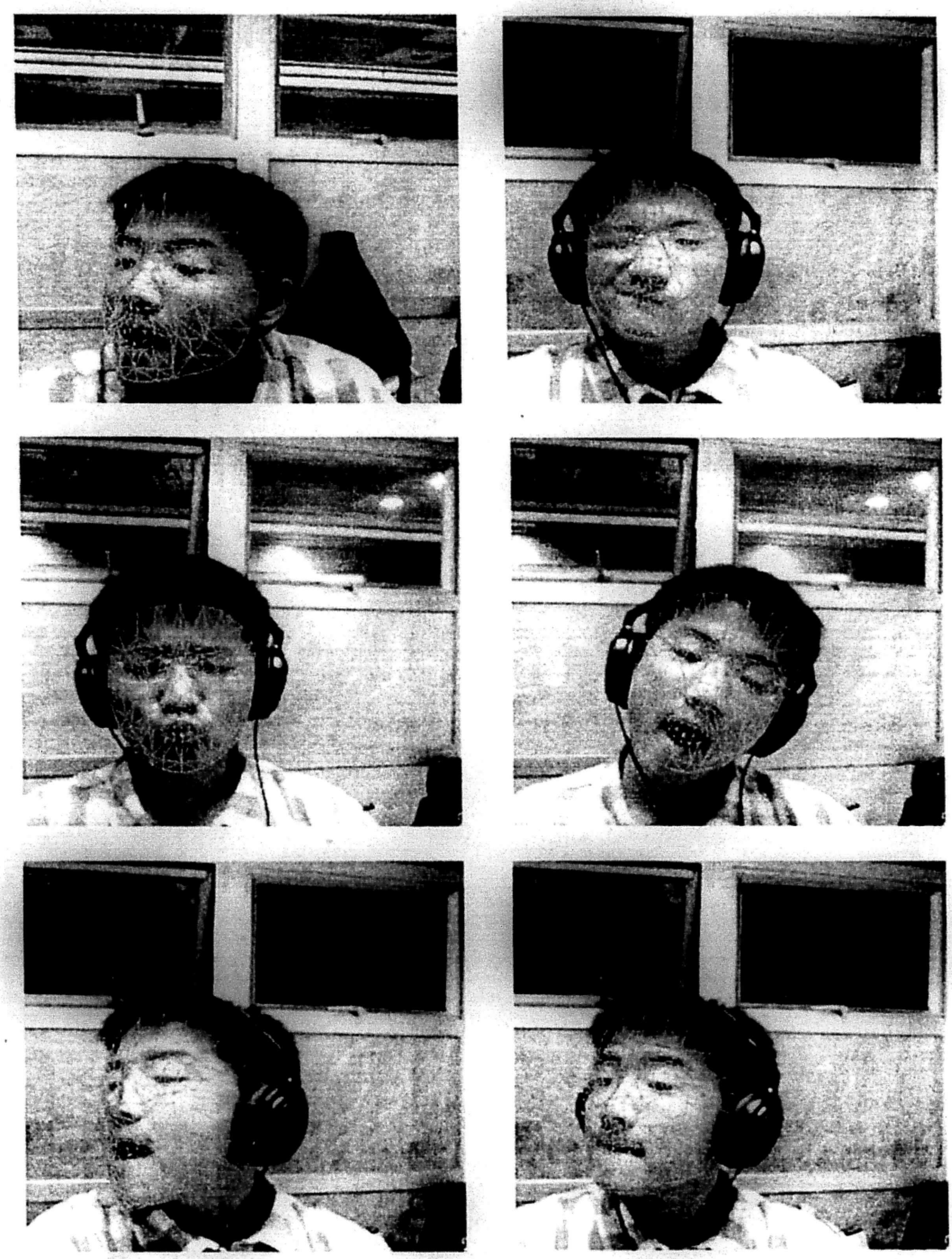


Figure 5.2: Nonrigid Face Tracking Results

model is different from the source one. Given the non-rigid 3D face tracking module, the 3D rigid motion (6 DOF of head motion) and expression motion are intrinsically decoupled. Transferring of the rigid 3D motion is straightforward, we can apply the 3D pose parameters from the source model directly to the target model. As for the non-rigid facial motion, we first extract the expression parameters from the face deformation and then utilize the expression parameters to control the target model's animation.Since our animation system is based on the MPEG-4 standard, we will give a brief introduction to the definitions related to the facial animation.

5.3.1 Motion Parameterization via MPEG-4 Encoding

MPEG-4 has a standard for 3D human facial animation. It includes the definition of two sets of parameters: Facial Definition Parameters (FDPs), and Facial Animation Parameters (FAPs). The FDPs define the shape and texture of the human face. It also specifies how the face mesh will deform according to the FAP values. The FAPs are account for the animation of facial model. MPEG-4 includes the common way to implement the facial animation based on the FAPs and FDPs. An advantage of the MPEG-4 based facial animation system is that little data are required to drive the facial animation system. Furthermore, it doesn't need much computing work, so it is suitable for real time animation on very low-band network.

FDP includes information about feature points position, and Face Def Tables (FDT), etc. Each domain of FDP includes the following 2 key items:

- FeaturePointsCoord: Specifies the coordinates of face feature points;

- FDT: Depicts how to use the FAPs to animate facial meshes.

In the standard of MPEG-4, there is a definition for 84 feature points in the FDP. The FAPs are a group of facial animation parameters. They represent a complete set of basic facial actions, and allow for the representation of most facial expressions. There are altogether 68 FAPs in the MPEG-4 standard, which can be divided into two groups. The first group contains two high-level FAPs: viseme FAP and expression FAP. We can express particular expression or viseme using the linear combination of the predefined basic expression FAPs or viseme FAPs. The other group contains 66 low-level FAPs that express motion of different specific regions on facial models. The function of the two high level FAPs is to express visemes and common expressions conveniently. The function performed by high level FAPs can also be done by low level FAPs. Since The low level FAPs can also express the complex and irregular expressions that can't be expressed by high level FAPs. In our thesis, we mainly deal with the low-level FAPs.

All the low-level FAPs are expressed in term of the Facial Animation Parameter Units (FAPUs). FAPUs are correspond to ratios of distances between some key facial features to a basic unit (1024 in our experiments). To be specific, there are altogether 6 FAPUs in the MPEG-4 standard: IRISD, ES, ENS, MNS, MW and AU. The definitions for FAPUs are as the following:

$$\begin{aligned} IRISD &= IRISD0/1024 \\ ES &= ES0/1024 \\ ENS &= ENS0/1024 \\ MNS &= MNS0/1024 \\ MW &= MW0/1024 \end{aligned} \tag{5.2}$$

The FAPUs depends on the facial model, and different facial models have different FAPUs. These units are defined in order to allow interpretation of the FAPs on any facial model in a consistent way. It means that a group of FAP represents the same expression on any facial model, thus making the FAPs universal. The physical meanings of FAPs are illustrated in Table. 5.1.

In the MPEG-4 based facial animation system, if we get the value of an FAP, we need to look up the FDT to get information about the control region

ID	Action	Min/Max Value	FAPU
3	open_jaw	0/1080	MNS
4	lower_t_midlip	-600/600	MNS
5	raise_b_midlip	-1860/1860	MNS
6	stretch_l_cornerlip	-600/600	MW
7	stretch_r_cornerlip	-600/600	MW
8	lower_t_lip_lm	-600/600	MNS
9	lower_t_lip_rm	-600/600	MNS
10	raise_b_lip_lm	-1860/1860	MNS
11	raise_b_lip_rm	-1860/1860	MNS
19	close_t_l_eyelid	-1080/1080	IRISD
20	close_t_l_eyelid	-1080/1080	IRISD
31	raise_l_I_eyebrow	-900/900	ENS
32	raise_r_I_eyebrow	-900/900	ENS
33	raise_l_m_eyebrow	-900/900	ENS
34	raise_r_m_eyebrow	-900/900	ENS
35	raise_l_o_eyebrow	-900/900	ENS
36	raise_r_o_eyebrow	-900/900	ENS
37	squeeze_l_eyebrow	-900/900	ES
38	squeeze_r_eyebrow	-900/900	ES
39	puff_l_cheek	-900/900	ES
40	puff_r_cheek	-900/900	ES
41	lift_l_cheek	-600/600	ENS
42	lift_r_cheek	-600/600	ENS
53	stretch_l_cornerlip_o	-600/600	MW
54	stretch_r_cornerlip_o	-600/600	MW
61	stretch_l_nose	-540/540	ENS
62	stretch_r_nose	-540/540	ENS
63	raise_nose	-680/680	ENS
64	bend_nose	-900/900	ENS

Table 5.1: FAP Values for Expression Synthesis

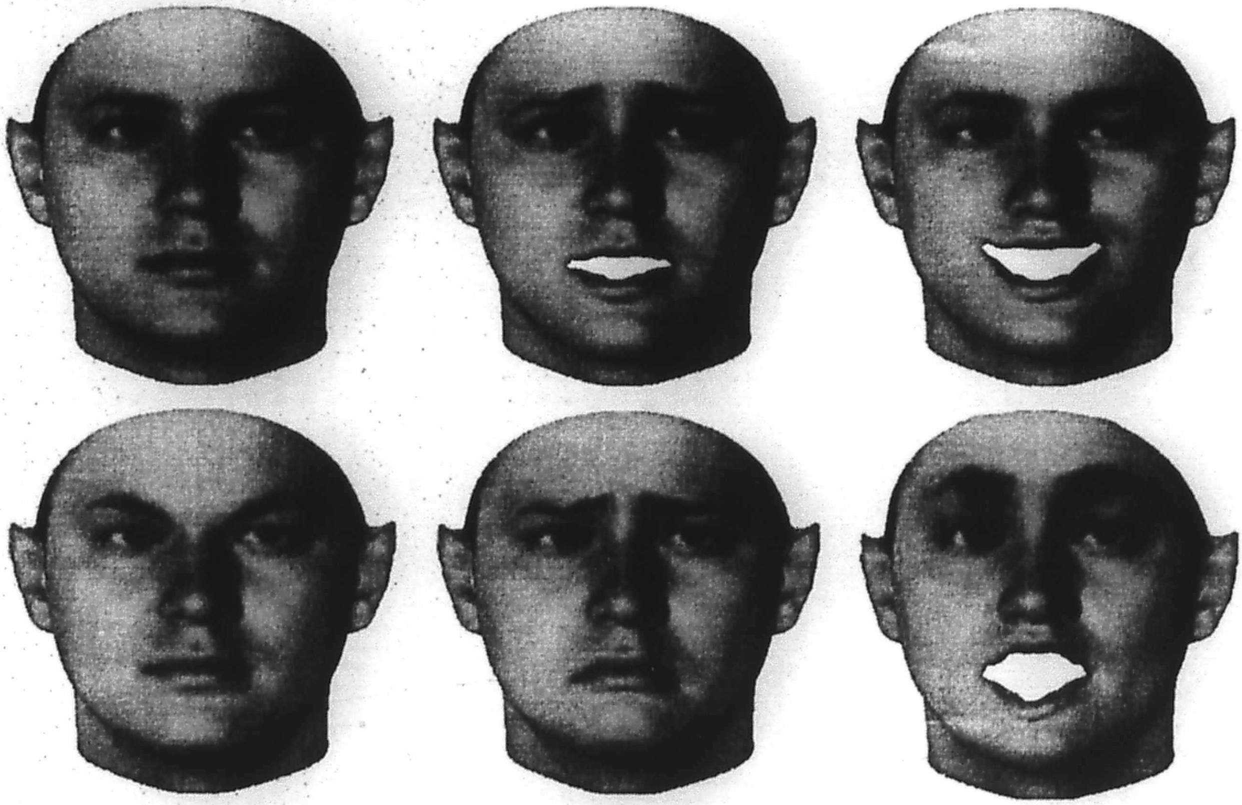


Figure 5.3: Avatar animation by FAP values

of the FAP and the three dimensional motion factors of vertices within the control region to convert the FAP into facial animation. Obviously, FDT play an important role in an MPEG-4 based facial animation system. In our thesis, we use the predefined FDT for 3D face reconstruction in [42].

5.3.2 Transferring Expression Parameters

In order to model the expression variations from an animator, we assume that a generic expressional face shape is the sum of an identity shape and an expression deformation:$\mathbf{S}_{exp} = \mathbf{S}_{id} + \Delta\mathbf{S}$, where $\mathbf{S}_{id}$ and $\Delta\mathbf{S}$ hold respectively the face shape with neutral expression and the displacements of the vertices due to expression variation. In our work, $\mathbf{S}_{id}$ is obtained at the initialization stage when the input neutral face is registered with a deformable 3D model described in previous section. Meanwhile, $\Delta\mathbf{S}$ is expressed as a linear combination of

FAPs:

$$\Delta \mathbf{S} = \sum_{i=1}^{n} \mathbf{C}_i^{fap} \alpha^i. \tag{5.3}$$

Given the above equation, the recovery of expression parameters α is cast as a quadratic optimization problem, in which we need to minimize the difference between $\mathbf{C}_i^{fap}$ with the current motion vector $\Delta\mathbf{S}$. However, the unconstrained error minimization of linear deformation might incur unexpected distortion, as described in [19]. We take the constraints on expression parameters to a fixed region $[\alpha_l, \alpha_u]$ and the optimization problem has the following form of standard quadratic programming:

$$min\ ||\Delta S - \sum_{i=1}^{n} C_i^{fap} \alpha^i||^2 \tag{5.4}$$
$$w.r.t\ \alpha_l \leq \alpha \leq \alpha_u$$

which can be handled efficiently with the standard numerical techniques [27].

Once the expression parameters are recovered, the motion retargeting to a new face model becomes a simple task, the 3D pose parameters and FAPs are transferred to the target model for animation. With the decouple of the facial motion capture and motion retargeting, it is very easy to reuse the the same source videos on different models. Figure. 5.4 shows the typical animation results from different animators. More demo results are available on [109].

5.4 Conclusion

We present a method of performance driven 3D facial animation by using a combination of non-rigid 3D tracking and shape blending for target animation. Given an animation character as usual, the proposed algorithm allows any users in front of a camera to drive facial animation, rather than animate by hand. The approach is effective even when the source and target animation model have quite different shapes. Different from previous approaches where

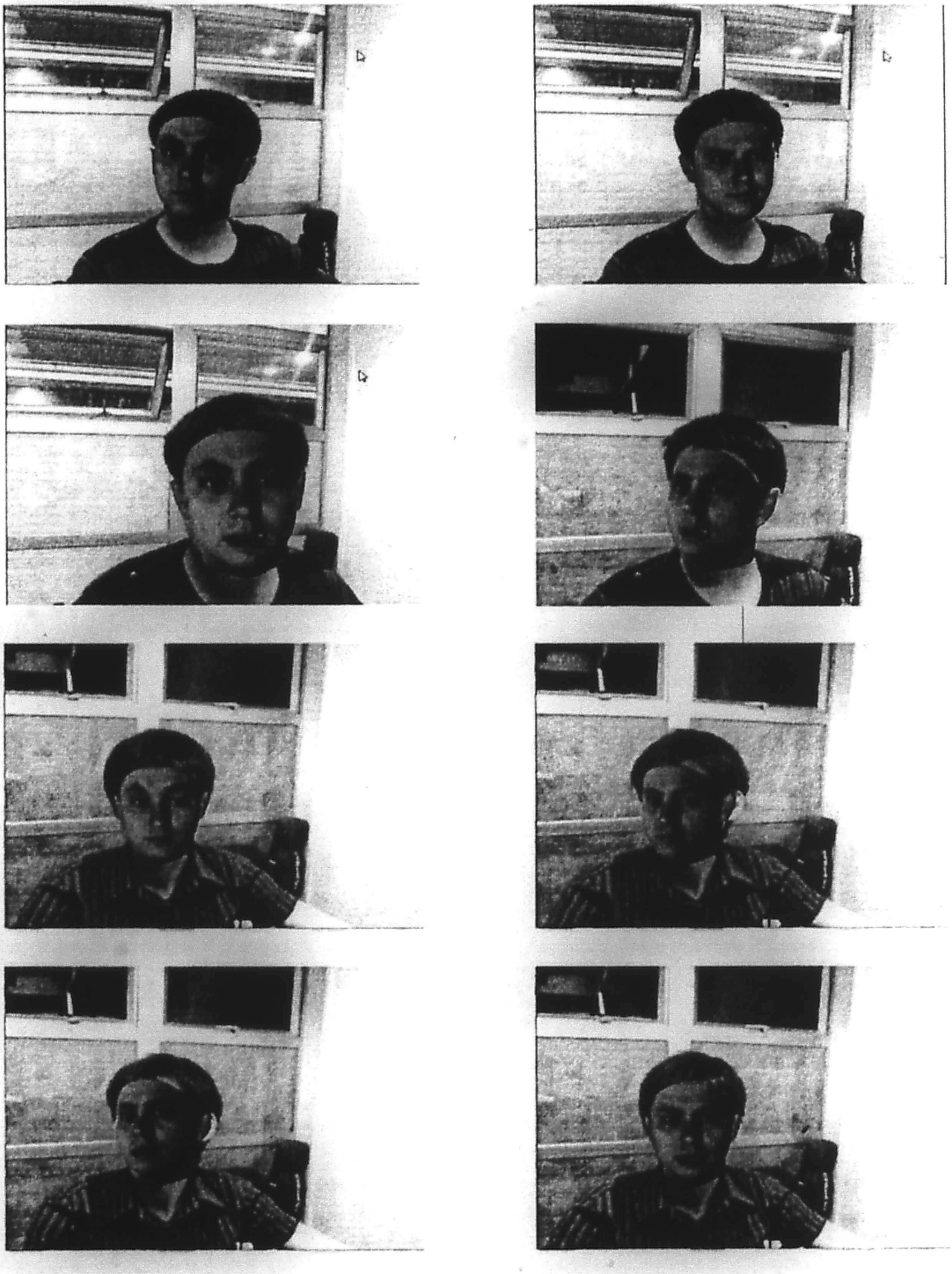


Figure 5.4: Animation Results

the animation is driven from sparse local features, our motion retargeting work is closer to the category of motion parameterizations, where the holistic 3D model's deformation is used for the high-level qualitative expressions parameters calculation.

Chapter 6

Conclusion And Future Work

In the first part of the thesis, we focus on the problem of three-dimensional face tracking, a challenging vision task that has many advantages in the fields of face recognition and performance driven facial animation. We develop a novel framework for 3D tracking of the non-rigid face deformation from a single camera. The challenging parts of the problem lies in the fact that 3D deformation parameter estimation becomes unstable when there are few reliable facial features correspondences. Meanwhile, in the real tracking scenarios there exists significant illumination change, motion blur or large pose variation. In order to extract more information of feature correspondences, the proposed framework integrates three types of features which discriminate face deformation across different views. The integration of these complementary features is important for robust estimation of the 3D parameters. In order to estimate the high dimensional 3D deformation parameters, we develop a hierarchical parameter estimation algorithm to robustly estimate both rigid and non-rigid 3D parameters. We show the importance of both features fusion and hierarchical parameter estimation for reliable tracking 3D face deformation.

The experiments of non-rigid tracking are performed on both real-world videos and live camera with low resolution input. To test the proposed algorithm on the videos in the indoor environment, we build a video database with

182 persons. For each identity in the database, a controlled motion is captured : the first 100 frames hold mainly frontal pose with neural expression; thereafter, one has around 600 frames in which no constraints are posed on the head motion and the expression variation except that the sequence starts from a frontal pose. Meanwhile, the video clips are captured in different environment and day-time. For the live camera device, the typical video resolution is set to 320 × 240. The difficulties for tracking in these videos stem from the out-of-plane motion, large expression change and numerous occlusions.

The proposed tracking algorithm works well for most cases with a fixed tracking configuration. However, there still exists some failure cases. In our experiments, most failure cases are due to the insufficient localization ability of the local feature models when side view faces are presented. We attribute the insufficient discrimination power of the local feature to the the synthesized training samples via 3D face reconstruction approach. The facial texture of synthesized samples are obtained from frontal images, in which the self-occlusion part is interpolated from the visible part, which might differ from the ground truth.

In the second part of the thesis, we put our focus on the topic of video based face recognition. A framework for face recognition in real-world noisy videos is proposed on the basis of 3D deformable face tracking. The difficulty of video face recognition tasks lies in the challenging appearance variations in real-world videos due to motion blur, large head rotation, occlusion, illumination change and significant image noise. To achieve the goal of accurate localization of faces in videos, the proposed 3D tracking algorithm makes good use of 3D face shape priors, local appearance model of major facial features, face silhouette and online feature matches across video frames. As opposed to the state-of-the-art video face recognition algorithm which relies on discriminative appearance model to classify face images into different views, our 3D tracking algorithm can directly estimate face pose for a view-based face recognition algorithm, which is more robust to outliers with comparison to the clustering based strategy. Furthermore, the proposed 3D tracking algorithm has a probabilistic form and can provide confidence measurement for the tracking result, which can be used to select high confidence frames. Such a frame selection strategy ensures the robustness of face recognition process.

At the recognition stage, three types of features are used for face representation: Regularized LDA, LE and sparse representation. To integrate all the recognition results from single frames, simple decision combination schemes are used including voting and metric fusion. With the proposed recognition framework, a high recognition rate can be achieved even with a simple feature descriptor and classifier (Regularized LDA + NN). Extensive experiments carried out on the real world noisy videos from YouTube demonstrate the significant improvement achieved with our approach: the accurate recognition rate reaches 79.8% for 46 persons, which outperforms the best reported results on the same data set (71.24%) with 35 persons.

In the last part of the thesis, a performance driven 3D face animation system is proposed, where the realism facial animation is achieved on a frame-by-frame basis. With the usage of low cost web camera, a captured performance is retargeted on to a morphable 3D face model based on a semantic correspondence between the facial landmarks and the 3D face model. The resulting facial animation reveals a high level of realism by combining the high resolution of a 3D morphable model with the high temporal accuracy of captured motion data that accounts for subtle facial movements with sparse measurements. Though we have made encouraging progress in the topic of non-rigid face tracking, it is far from the end of the road. A lot of work can be done to make further improvement.

6.1 Future Work

No research is ever truly complete, and the work described in this thesis is no exception. The purpose of this section is to briefly mention some of the extensions that could enhance the performance of the non-rigid tracking and the video based face recognition. Some of the extensions are incremental, while others would be entirely new ground.

6.1.1 Automatic Discriminative Components Selection for Multi-view Face Alignment

As described in Chapter. 3, There are still some failure cases for the proposed tracking framework, especially when the large view switch and expression change occurs. The failure is mainly due to the insufficient localization ability of the local feature models when side view faces are presented. We attribute the insufficient discrimination power of the local feature to the the synthesized training samples via 3D face reconstruction approach. Therefore, we need to develop an approach for tracking recovery and re-initialization via an automatic face registration technique, which is independent from the tracking process. Meanwhile, the multi-view case should be considered since the tracking could be failed in arbitrary view.

For a typical automatic face alignment algorithm, the initial alignment configuration is determined by a face detector. Due to the large intra-class variation for face detection dataset, the holistic apprqach by treating the whole face as one object may drift from its true location in some cases, especially when the side view samples are presented. When the initialization location is out of the scope of alignments convergence, they can not search the truth position.

A natural solution to this problem is to use facial component detectors for a refined shape localization. In [56], Liang *et.al.* try to solve the alignment initialization problem via using a set of facial component detectors and direction classifiers for refined initial guess of the facial components position. The key idea of their work is a component level shape localization. The traditional ASM style alignment is performed on the base of coarse level localization to achieve high precision. This work has its deficiency that the components location are manually selected, they are mainly used for near frontal face alignment. When multi-view case is considered, a multi-view component detectors need to be trained, which have bad generalization abilities. Therefore, we try to tackle the multi-view facial registration problem by automatically selecting a set of components for face localization, where the components are adaptively selected with respect to the view of face. Given the set of training face images with face locations, we select the most discriminative components for localization via the rankboosting scheme. The ensemble of these components are used for a more precise face location in arbitrary view.

6.1.2 Video Based Recognition via Spatial Coherence Sparse Representation

One property of faces from image sequences is that their tendency to be spatially coherent. This strong source of regularity has not been explicitly leverage in the work present in Chapter. 4. However, it should be noticed that the spatial coherent of the facial appearance is widely used constraint in most manifold learning algorithms for face recognition. Most significant gains might be achieved by enforcing spatial coherence on the query videos at the stage of sparse representation feature extraction. For example, if we assume the appearance of an input videos lies in a low-dimensional space, an appearance manifold could be learnt via [52]. For each subspace approximation of the appearance manifold, we can enforce the constraint that the coefficients of the samples lies in the same subspace will be similar. An potential advantage of spatial coherence would be to suppress the diversity of the sparse coefficients over the whole image sequence, the sparse representation of the whole video is solved globally to pursue the goal of robust recognition.

Bibliography

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. *European Conference on Computer Vision*, 2004.

[2] O. Arandjelovic and A. Zisserman. Automatic Face Recognition for Film Character Retrieval in Feature-Length Films. *IEEE Conference on Computer Vision and Pattern Recognition*, 1, 2005.

[3] M. Bartlett, J. Hager, P. Ekman, and T. Sejnowski. Measuring facial expressions by computer image analysis. *Psychophysiology*, 36, 1999.

[4] M. Bartlett, J. Movellan, and T. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13, 2002.

[5] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. 13, 1996.

[6] M. Black. Robust incremental optical flow. *Yale University, New Haven, CT*, 1992.

[7] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26, 1998.

[8] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.

[9] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. 2, 2000.

[10] C. Bregler and J. Malik. Tracking people with twists and exponential maps. 1998.

[11] T. Brox, B. Rosenhahn, D. Cremers, and H. Seidel. High accuracy optical flow serves 3-D pose tracking: exploiting contour and flow based constraints. *European Conference on Computer Vision*, 6, 2006.

[12] A. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. 2, 2005.

[13] C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2, 1998.

[14] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face Recognition with Learning-based Descriptor. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[15] J. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. 2003.

[16] R. Chellappa, C. Wilson, S. Sirohey, et al. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83, 1995.

[17] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern recognition*, 33, 2000.

[18] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43, 2001.

[19] E. Chuang and C. Bregler. Performance driven facial animation using blendshape interpolation. *Computer Science Technical Report, Stanford University*, 2002.

[20] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *European Conference on Computer Vision*, 1998.

[21] T. Cootes, C. Taylor, D. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61, 1995.

[22] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models withapplications to human face shape and motion estimation. 1996.

[23] D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15, 1995.

[24] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977.

[25] D. Donoho. For most large underdetermined systems of linear equations the minimal l1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59, 2006.

[26] P. Eisert and B. Girod. Analyzing facial expressions for virtual conferencing. *IEEE Computer Graphics and Applications*, 18, 1998.

[27] Q. P. Ekaterina, E. Kostina, and O. Kostyukova. A primal-dual active-set method for convex. 2003.

[28] P. Ekman and W. Friesen. Facial action coding system: A technique for the measurement of facial movement. 1978.

[29] I. Essa and A. Pentland. Coding, analysis, interpretation, and recognition of facialexpressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1997.

[30] I. Essa, A. Pentland, P. Sect, and C. MIT. A vision system for observing and extracting facial actionparameters. 1994.

[31] M. Everingham and A. Zisserman. Identifying individuals in video by combining generative and discriminative head models. *IEEE International Conference on Computer Vision*, 2005.

[32] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma. Learning the structure of manifolds using random projections. 2007.

[33] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research*, 4, 2003.

[34] D. Gennery. Visual tracking of known three-dimensional objects. *International Journal of Computer Vision*, 7, 1992.

[35] A. Goldstein, L. Harmon, and A. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59, 1971.

[36] L. Gu and T. Kanade. 3D Alignment of Face in a Single Image. *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[37] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. 1998.

[38] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry andillumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 1998.

[39] P. Hallinan. Recognizing human eyes. 1570, 1991.

[40] C. Harris and M. Stephens. A combined corner and edge detector. 15, 1988.

[41] A. Hill, T. Cootes, and C. Taylor. Active Shape Models and the shape approximation problem. *Image and Vision Computing*, 14, 1996.

[42] Y. Hu, D. Jiang, S. Yan, L. Zhang, et al. Automatic 3D reconstruction for face recognition. *IEEE Conference on Automatic Face and Gesture Recognition*, 2004.

[43] G. Hua and A. Akbarzadeh. A Robust Elastic and Partial Matching Metric for Face Recognition. *IEEE International Conference on Computer Vision*, 2009.

[44] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report*, 57, 2007.

[45] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[46] P. Joshi, W. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. 2005.

[47] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1, 1988.

[48] M. Kimg, S. Kumar, V. Pavlovic, and H. Rowley. Face Tracking and Recognition with Visual Constraints in Real-World Videos. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[49] B. Knight and A. Johnston. The role of movement in face recognition. *Visual Cognition*, 4, 1997.

[50] M. La Cascia and S. Sclaroff. Fast, reliable head tracking under varying illumination. 1, 1999.

[51] S. Lawrence, C. Giles, A. Tsoi, and A. Back. Face recognition: A convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8, 1997.

[52] K. Lee, J. Ho, M. Yang, and D. Kriegman. Video-Based Face Recognition Using Probabilistic Appearance Manifolds. *IEEE Conference on Computer Vision and Pattern Recognition*, 1, 2003.

[53] K. Lee, J. Ho, M. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, 99, 2005.

[54] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. 2004.

[55] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. 1, 2001.

[56] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search. 1, 2008.

[57] D. Lin and X. Tang. Recognize high resolution faces: From macrocosm to microcosm. 2, 2006.

[58] C. Liu and H. Wechsler. Enhanced Fisher Linear Discriminant Models for Face Recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[59] X. Liu. Generic face alignment using boosted appearance model. 2007.

[60] Z. Liu and Z. Zhang. Robust head motion computation by taking advantage of physicalproperties. 2000.

[61] D. Lowe. Robust model-based motion tracking through the integration of search and estimation. *International Journal of Computer Vision*, 8, 1992.

[62] D. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60, 2004.

[63] A. Martinez and R. Benavente. The AR face database. *CVC Technical Reportno*, 24, 1998.

[64] A. Martinez and A. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, 2001.

[65] I. Matthews and S. Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60, 2004.

[66] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The Extended M2VTS Database. *Second International Conference on Audio and Video-based Biometric Person Authentication*, 626, 1999.

[67] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005.

[68] H. Moon and P. Phillips. Computational and performance aspects of PCA-based face-recognition algorithms. *PERCEPTION-LONDON-*, 30, 2001.

[69] J. Noh and U. Neumann. Expression cloning. 2001.

[70] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray scale and rotation invariant texture analysis with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 2002.

[71] F. Parke. Computer generated animation of faces. 1972.

[72] N. Peter, P. João, and J. David. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[73] P. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, M. Bone, and A. DARPA. Face recognition vendor test 2002. 2003.

[74] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5), 1998.

[75] F. Pighin, R. Szeliski, and D. Salesin. Modeling and animating realistic faces from images. *International Journal of Computer Vision*, 50, 2002.

[76] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, 1994.

[77] J. Stallkamp, H. Ekenel, and R. Stiefelhagen. Video-based Face Recognition on Real-World Data. *IEEE International Conference on Computer Vision*, 2007.

[78] Y. Taigman, L. Wolf, T. Hassner, and I. Tel-Aviv. Multiple one-shots for utilizing class label information. 2009.

[79] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Analysis and Modeling of Faces and Gestures*, 2007.

[80] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 1996.

[81] P. Torr and C. Davidson. IMPSAC: Synthesis of importance sampling and random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 2003.

[82] A. Torralba, K. Murphy, and W. Freeman. The MIT CSAIL Database of objects and scenes. 2003.

[83] L. Torresani, D. Yang, E. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. 1, 2001.

[84] M. Turk and A. Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3, 1991.

[85] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3-D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[86] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[87] T. Vetter. Synthesis of novel views from a single face image. *International Journal of Computer Vision*, 28, 1998.

[88] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple. 2001.

[89] C. Vogler, Z. Li, A. Kanaujia, S. Goldenstein, and D. Metaxas. The Best of Both Worlds: Combining 3D Deformable Models with Active Shape Models. *IEEE International Conference on Computer Vision*, 2007.

[90] Q. Wang, W. Zhang, X. Tang, and H. Shum. Real-Time Bayesian 3-D Pose Tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 16, 2006.

[91] X. Wang and X. Tang. Unified subspace analysis for face recognition. *IEEE International Conference on Computer Vision*, 2003.

[92] X. Wang and X. Tang. Dual-space linear discriminant analysis for face recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[93] L. Williams. Performance-driven facial animation. 1990.

[94] L. Williams. Performance-driven facial animation. 2006.

[95] L. Wiskott, J. Fellous, N. Kruger, and C. Von der Malsburg. Face recognition by elastic bunch graph matching. pages 456–463.

[96] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. 2008.

[97] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008.

[98] H. Wu, X. Liu, and G. Doretto. Face alignment via boosted ranking model.

[99] C. Wylie, G. Romney, D. Evans, and A. Erdahl. Half-tone perspective drawings by computer. 1967.

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

[101] J. Xiao, T. Kanade, and J. Cohn. Robust full-motion recovery of head by dynamic templates andre-registration techniques. 2002.

[102] R. Xiao, L. Zhu, and H. Zhang. Boosting chain learning for object detection. *IEEE International Conference on Computer Vision*, 2003.

[103] L. Xin, Q. Wang, J. Tao, X. Tang, T. Tan, and H. Shum. Automatic 3D Face Modeling from Video. *IEEE International Conference on Computer Vision*, 2005.

[104] Y. Xu and A. Roy-Chowdhury. Integrating the Effects of Motion, Illumination and Structure in Video Sequences. *IEEE International Conference on Computer Vision*, 2, 2005.

[105] S. Yan, M. Li, H. Zhang, and Q. Cheng. Ranking prior likelihood distributions for Bayesian shape localization framework. *IEEE International Conference on Computer Vision*, 2003.

[106] J. Zhang, S. Zhou, D. Comaniciu, and L. McMillan. Discriminative learning for deformable shape segmentation: A comparative study. *ECCV 2008*, 2008.

[107] L. Zhang, R. Chu, S. Xiang, S. Liao, and S. Li. Face detection based on multi-block lbp representation. *Advances in Biometrics*, 2007.

[108] W. Zhang. Deformable face tracking demos. Website. `http://mmlab.ie.cuhk.edu.hk/Demos/NRF.zip`.

[109] W. Zhang. Performance driven animation demos. Website. `http://mmlab.ie.cuhk.edu.hk/Demos/FaceAnimation_Demos.rar`.

[110] W. Zhang, Q. Wang, and X. Tang. Real Time Feature Based 3-D Deformable Face Tracking. *European Conference on Computer Vision*, 2008.

[111] Z. Zhang, Z. Liu, D. Adler, M. Cohen, E. Hanson, and Y. Shan. Robust and rapid generation of animated faces from video images: A model-based modeling approach. *International Journal of Computer Vision*, 58, 2004.

[112] P. Zhao and B. Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7, 2006.

[113] W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. *IEEE Conference on Automatic Face and Gesture Recognition*, 1998.

[114] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4), 2003.

[115] Y. Zhou, W. Zhang, X. Tang, and H. Shum. A Bayesian Mixture Model for Multi-view Face Alignment. *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.