Summer 2011

The Tracking Performance of Distributed Recoverable Flight Control Systems Subject to High Intensity Radiated Fields

Rui Wang
Old Dominion University

CREATION OF LARGE SCALE FACE DATASET

USING SINGLE TRAINING IMAGE

by

Qun Wang
B.S. June 2000, Shandong University, China
M.S. June 2004, Xidian University, China

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL AND COMPUTER ENGINEERING

OLD DOMINION UNIVERSITY
May 2012

Approved by:

_____
Mohammad A. Karim (Director)

_____
Vijayan K. Asari (Co-Director)

_____
Jiang Li (Member)

_____
Yuzhong Shen (Member)

ABSTRACT

CREATION OF A LARGE SCALE FACE DATASET
USING SINGLE TRAINING IMAGE

Qun Wang
Old Dominion University, May 2012
Co-Directors: Dr. Mohammad A. Karim
Dr. Vijayan K. Asari

Face recognition (FR) has become one of the most successful applications of image analysis and understanding in computer vision. The learning-based model in FR is considered as one of the most favorable problem-solving methods to this issue, which leads to the requirement of large training data sets in order to achieve higher recognition accuracy. However, the availability of only a limited number of face images for training a FR system is always a common problem in practical applications. A new framework to create a face database from a single input image for training purposes is proposed in this dissertation research. The proposed method employs the integration of 3D Morphable Model (3DMM) and Differential Evolution (DE) algorithms. Benefitting from DE's successful performance, 3D face models can be created based on a single 2D image with respect to various illumination and pose contexts. An image deformation technique is also introduced to enhance the quality of synthesized images. The experimental results demonstrate that the proposed method is able to automatically create a virtual 3D face dataset from a single 2D image with high performance. Moreover the new dataset is capable of providing large number of face images equipped with abundant variations. The validation process shows that there is only an insignificant difference between the input image and the 2D face image projected by the 3D model. Research work is progressing to consider a nonlinear manifold learning methodology to embed the

synthetically created dataset of an individual so that a test image of the person will be attracted to the respective manifold for accurate recognition.

This dissertation is dedicated to my father and mother

ACKNOWLEDGMENTS

I would like to express my gratitude to Dr. Vijayan K. Asari and Dr. Mohammad A. Karim, for their valuable academic guidance, encouragement, help, support and patience over the years of my graduate study in ODU.

I also would like to thank Dr. Jiang Li for his continuous advising and tutoring on my research work.

I would like to thank Dr. Yuzhong Shen for his suggestions that speeded up my research progress.

I would also like to acknowledge the Electrical and Computer Engineering Department, for its support all these years.

I would like to thank all the colleagues I worked with in the Vision Lab for their cooperation to my research. Great thanks to all my friends, Dr. Liu Wang, Dr. Jie Zhuang, Dr. Xiao Yan Sun, Ms. Gail Parker and all others for their help.

Finally, I would like to dedicate my work to my beloved parents. Thanks for their everlasting support and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

Figure   Page

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Overview

1.1.1 Introduction

During the last 40 years, researchers have achieved significant progress in the field of face recognition (FR) [1]. Especially in the recent decade, a number of substantial advances have been developed with the emergence of evaluation systems for FR. The early research utilized no more than 10 images for the system validation. It was not until middle 1990s that methods were proposed and tested on large scale face datasets. Thereafter the idea aiming at establishing a common evaluation system containing thousands of diversified face images became popular.

In 1994, Y. Moses et al. revealed that "the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity" [2]. As shown in Figures 1.1, 1.2 and 1.3, face images taken from the same person, with the same facial expression, lead to dissimilar appearance with great difference when light source directions or viewpoints vary[3, 4].
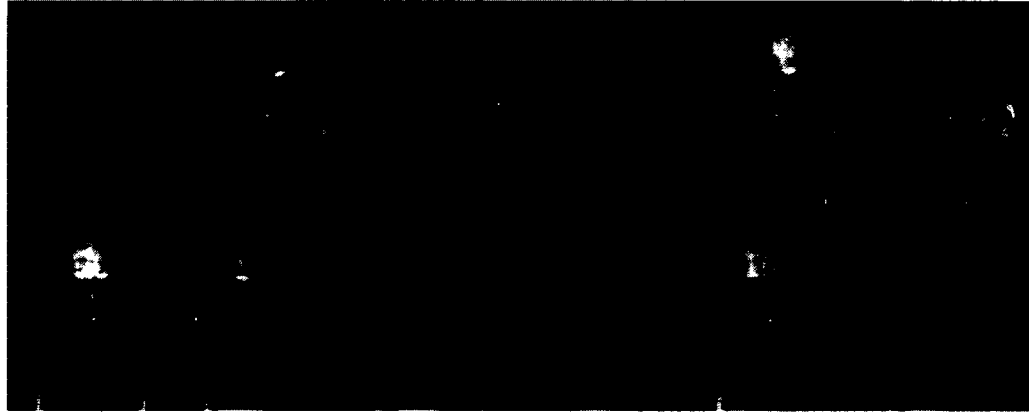
Fig. 1.1 Frontal face images taken under different controlled lighting conditions
(Courtesy of Yale Face Database B)
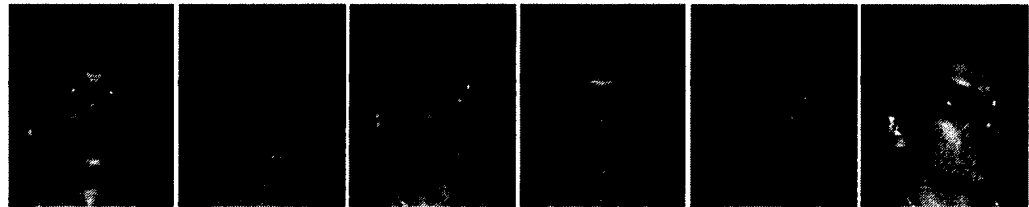
Fig. 1.2 Face images taken from different viewpoints
(Courtesy of Yale Face Database B)

Fig. 1.3 Face images taken from different subjects
(Courtesy of Yale Face Database B)

1.1.2 Public Face Databases

Machine learning plays a very important role in computer vision (CV). It has become

common sense that most of solutions in CV, such as scene matching, 3D reconstruction,

classification and segmentation, can be approximated by using machine learning

algorithms. In 1983 and 1986, Dietterich et al. [5, 6] pointed out that "pure empirical

learning systems require a set of examples that sufficiently cover the problem space in

order for the algorithm to produce a description of the problem class." Based on this common awareness, researchers have developed a large number of face databases to meet the demands for data training purposes.

In the following section, we present four representative face databases (refer to Figure 1.4-1.8). These databases have their own characteristics and are different from each other. Some of them involve face images taken under scenarios of various controlled or uncontrolled lighting sources, and combinations thereof. Some of them provide expression variations beside neutral faces and pose variations. The number of subjects of each database varies from 10-20 to more than 300 [7].

(I)   AR Database
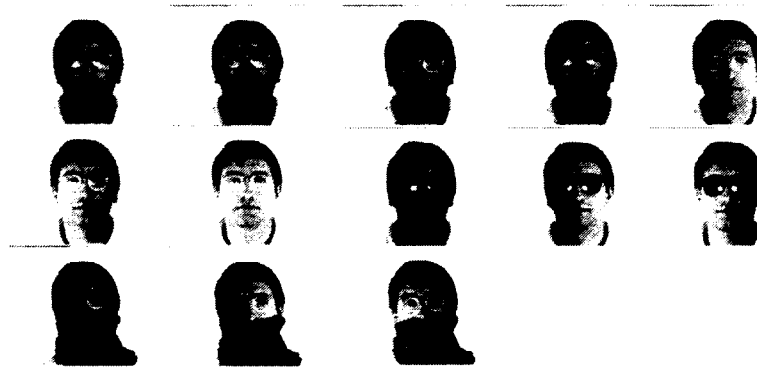
Fig. 1.4 Images from AR Database. The conditions are (from left to right, top to bottom) neutral, smile, anger, scream, right light on, left light on, both lights on, sun glasses, sun glasses/right light, sun glasses/left light, scarf, scarf/right light, scarf/left light
(Courtesy of AR Database)

(II)   BANCA

Fig. 1.5 Images from BANCA Database: (Top) taken under controlled conditions,
(Middle) degraded image quality, (Bottom) adverse environment conditions.
(Courtesy of BANCA Database)

(III)   CMU – PIE Database
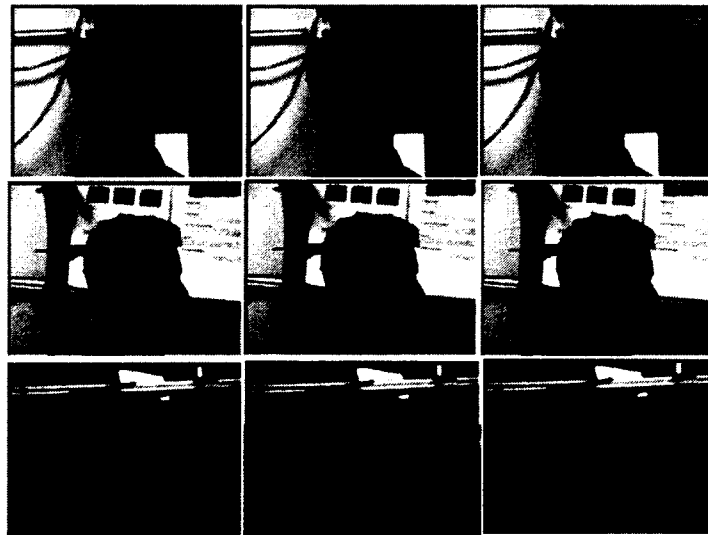
Fig. 1.6 Images from CMU-PIE Database with pose and expression variation
(Courtesy of CMU-PIE Database)

Fig. 1.7 Images from CMU-PIE Database with pose and illumination variation
(Courtesy of CMU-PIE Database) [8]

(IV)   Yale Face Database B

Fig. 1.8 Frontal images taken under various lighting conditions
(Courtesy of Yale Face Database B)

1.2 Scope of the Dissertation Research

1.2.1 Motivation

Face recognition (FR) has become one of the most successful applications of image analysis and understanding in computer vision. In the last decades, researchers endeavored to establish FR systems based on the computational conception, which demonstrates its capability comparable to that of human perception system. The

commonly used representation of the facial appearance is obtained in the form of rasterisation, e.g. still intensity images or sequential images. Nevertheless, the potential obstacles to the efforts are exponentially increased not only by the complexity of facial appearance itself but also by illumination conditions, distance and perspective to the viewer and lens distortions, as well as the noises during the acquisition.

Among all the aforementioned disadvantages, pose and illumination have been identified as the major challenges in FR[1]. The nature of facial 3D structure leads to the fact that there is a large chance for part of the key facial features to be diminished in the shadows cast by direct lighting[9]. It has been experimentally proved that differences in appearances induced by the illumination are larger than those between individuals. The authors give their experimental results and explanation of the effects of changing illumination conditions to the performances of certain kinds of face recognition methods. They conclude in the paper that all the face recognition approaches they put into experiments, such as edge maps, image intensity derivatives and images convolved with 2D Gabor-like filters, can not provide satisfactory results by themselves although these methods are designed to avoid sensitivities from illumination variations. In addition, the same conclusions have also been drawn from experimental results under different poses and different expressions conditions.

The learning-based model is considered to be one of the most favorable problem-solving methods for this issue, which thus leads to the requirement of large training data sets in order for higher recognition accuracy. However, the circumstances of limited number of training images are a common problem. Consequentially, it makes sense to propose a scheme of creating face databases consisting of training images equipped with extra

variations based on the small set of candidate images. The generated database should then be capable of providing continuous face sequences of wide range of lighting and pose variations to the training set of face recognition framework.

Real Image   3D Face Model Generator   Generated Image Database of Virtual Faces
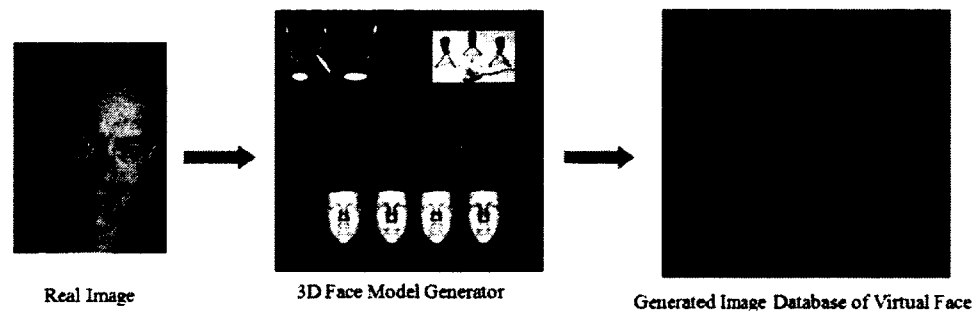
Fig. 1.9 Illustration of motivation of the creation of the large database from single face image

1.2.2 Research Feasibility

As aforesaid, the idea of creating a new database stems from the fact of *insufficient* information provided by the training resources. So acquiring *prior* knowledge appears to be an instinctive proposition. The *intrinsic* patterns and structures of the image exemplars, on which the *extrinsic* features are imposed, make it possible to disclose the distinctive features of certain class of objects. On the other hand, if the major characteristics of the specific class of objects, e.g. human faces and human bodies, are extractable, the assumption of producing new instances of the same class is valid.

The model-based method in computer vision was invented to enhance the capability of the system for image understanding and interpretation. Particularly, the emergence of statistical-based method provides an opportunity to depict the specific class of object in a generic framework that is represented in the form of a set of parameters. The method is

fitting the model to the given images and/or the associated features, which is basically a tuning-up process of the model parameters to the test images by producing its synthesized counterparts. This fact results in an explicit procedure of creating new instances of the specified class of objects. Figure 1.9 illustrates the basic concept of the scheme mentioned above.

1.2.3 Contributions

The first contribution of this dissertation is to propose a novel method for the 3D face synthesis and reconstruction by using a simple and efficient global optimizer. A 3D-2D matching algorithm, which employs the integration of the 3D morphable model (3DMM) and the differential evolution (DE) algorithm, is addressed. In 3DMM, the estimation process of fitting shape and texture information into 2D images is considered as the problem of searching for the global minimum in a high dimensional feature space, in which optimization is apt to have local convergence. Unlike the traditional scheme used in 3DMM, DE appears to be robust against stagnation in local minima and sensitiveness to initial values in face reconstruction. Benefitting from DE's successful performance, 3D face models can be created based on a single 2D image with respect to various illumination and pose contexts. The experimental results demonstrate that we are able to automatically create a virtual 3D face from single 2D image with high performance. The validation process shows that there is only an insignificant difference between the input image and the 2D face image projected by the 3D model.

Another contribution presented in this dissertation originates from moving our concentration from statistical-based methods to appearance-based methods. In order to

seek real image quality, we utilize backward texture mapping to operate transformations from the image space to the texture space. The key point is we tentatively borrow the idea from the image warping (e.g. Moving Least Square deformation) and integrate it in our work. The synthesized images demonstrate that the proposed approach is convenient, easy to use and significantly improves the final results.

1.2.4 Organization

In this dissertation, Chapter 2 provides an overview of the proposed framework. Chapter 3 focuses the relevant literature review and reports on the statistical 3D facial representation. Some discussions about key factors pertaining to the internal mechanisms associated with our models are also given. In Chapter 4, the detailed physical models are presented. The approaches and simulation scheme used in this dissertation research are described and explained in detail. Next, Chapter 5 shows the results obtained based on the models of Chapter 3 and Chapter 4. We also provide the validation process to the experimental results. In Chapter 6, we address a complementary scheme to enhance to result obtained from Chapter 5. A summary and conclusion of the present research along with the discussion on future works are given in Chapter 7.

CHAPTER 2

SYSTEM INTRODUCTION AND OVERVIEW

2.1 Introduction

The proposed system is a generic framework for the research topic of the creation of large scale face database from single training image. In other words, given a 2D input face image, the framework is able to synthesize a new face dataset with illumination and pose variations. Inspired by the realistic scene created by computer graphics technology, the system builds its problem-solving capacity upon the concept of 3D face models reconstruction as well as the 3D scene rendering. It is a systematic and nontrivial task, especially for the 3D face reconstruction, which requires the involvement of different technologies in machine learning, computer vision, image processing and computer graphics.

The framework deploys its solution into several modules, which are 3D model generation, 3D model enhancement, and 3D model rendering respectively. In the following sections, we will discuss each module and the associated units within them. The diagram (Figure 2.1) of the whole framework is shown at the end of this chapter.

2.2 3D Model Generation

The module of 3D model generation is the core of the whole system. It is designed to reconstruct the 3D face model from the input image. It consists of five parts, e.g. unit of

image processing, 3D Morphable model, a set of training weights, an optimizer and the cost function. The basic idea of the model generation is fitting 3D face model to the 2D input image by tuning up a set of parameters that are capable of handling the face model. The system takes iterations to execute the 3D-2D matching operations that are driven by the optimizer circulating through 3DMM and the cost function to approximate toward the optimal solution.

2.2.1 3DMM Unit

The 3DMM unit is based on a statistical-based model, which contains prior knowledge of 3D human faces, such as shape and texture representations of the mean face model as well as their corresponding principal components. It also manages the algorithms of dealing with the training weights, by which the 3D new face model is generated and projected onto the 2D image plane.

2.2.2 Optimizer

The implementation of 3D face reconstructions in our case is a nonlinear problem in which a large number of training weights complicates the situation. A global optimizer, named Differential Evolution (DE), is included in attempt to find the optimal combination of the training set. DE is a genetic algorithm-like method that is known for its excellent performance of global optimization. DE seeks the global optimum in an iterative working manner until the iteration is stopped by the cost function.

2.2.3 Cost Function

The cost function provides a quantitative description of the result came from 3DMM by indicating its distance to the input image. Only if the distance falls below the threshold or certain criteria is reached, the cost function constantly informs the optimizer to locate a better solution. The cost function represents itself in different forms with respect to different training phases in the estimation process. For example, the Training Phase I is executing rigid registration by measuring the distance between feature points of 3D object f_{3d} and that of 2D image f_{2d}. The feature point sets are manually selected before the training starts, which will be explained in detail in the Chapter 5. While in the Training Phase II, the cost function operates non-rigid registration and 3D-2D matching between the 3D morphable model and the input image. In our experiments, the cost function is designed on basis of 2-norm Euclidean distance function, also known as the function of Mean Square Error (MSE). There are multiple choices of distance functions that are available for the cost function, but we take MSE into consideration based on the tradeoff between the system performance and time consumption.

2.2.4 Training Weights

The set of training weights is a combination of parameters from different domains, e.g. Eigen space, color space and 3D Cartesian space. Literally speaking, they determine the success of the face reconstruction. During the process of optimization, the weight set with the best result in previous iterations will be maintained, or updated by a better result if there is any.

2.2.5 Image Processing Unit

3DMM is somehow incompatible with noise and blur. However, facial appearances bring challenges to the face model reconstruction due to the reason that they are instantiated based on images that will inevitably suffer from noise. The image processing unit is designed to tackle the disturbance caused by noise in the reconstruction process. At the same time, according to our experiments, it is critical and beneficial for the 3DMM to know the edge and outstanding information of the input image. Image processing unit is in charge of filtering out the unwelcome noise but retaining possession of the useful edge, shadow and lighting information at the same time.

2.3 3D Model Enhancement

The model enhancement module is set to reconstruct an approximated facial appearance to its ground truth by backward texture mapping. The new 3D face from the model generation module is augmented on its texture appearance. The module transforms the input image from the image plane to the texture plane and assigns it to the shape surface of the 3D face. The transformation of the input image to the texture plane demands for annotation landmarks on the image as the control points. Hence a user interface is also needed.

2.4 3D Model Rendering

The 3D rendering module is basically built upon the computer graphics technology. Since our goal is to create a large scale database, rendering the new face model in a 3D context

is an instinctive thought. Two sets of parameters control the output for new dataset, in which one is the lighting control and the pose control the other. The computational lighting model, accompanied by the pose calculation and the user definitions on both sides, ensures the diversification of new face images for the dataset creation. The overview diagram of the entire framework is shown in Figure 2.1, which is demonstrated on the next page.

3D MODEL GENERATION

3D Cartesian Space

$W_{Rotation}$
$W_{Translation}$
$W_{Scaling}$

Eigen Space

W_{Shape}
$W_{Texture}$

Color Space

W_{Hue}
$W_{Saturation}$
$W_{Intensity}$

W_{model} TRAINING WEIGHTS

INPUT IMAGE

IMAGE PROCESSING

3D MORPHABLE MODEL

OPTIMIZER
Differential Evolution

COST FUNCTION

3D MODEL ENHANCEMENET

IMAGE -> TEXTURE TRANSFORMATION

3D FACE MODEL

ENHANCED 3D FACE MODEL

USER INTERFACE

LIGHTING CONTROL

POSE CONTROL

3D FACE MODEL

3D MODEL RENDERING

2D FACE DATASET

Fig. 2.1 Overview diagram of the entire framework

CHAPTER 3

STATISTICAL 3D FACE REPRESENTATION

3.1 Background and Previous Works

3.1.1 Overview of Statistical-based Methods

The statistical-based method is a successful model to capture image patterns. A number of models have been addressed over the past years. It would be quite elaborate to cover all the papers in this area, so only a small number of milestones or breakthroughs will be mentioned in the following short review.

The first notable paper was published in 1995 written by T.F. Cootes, et al., who was inspired by an extraordinary paper [10] proposed by Kass, Witkin and Terzopoulos. The concentration of the research was originally directed toward the shape modeling. Active Shape Model (ASM) learns through observations, giving the model flexibility, robustness and specificity as the model only can synthesize plausible instances with respect to the observations [11]. Figure 3.1 shows the basic application of AAM in hand matching.

On the basis of ASM, T.F. Cootes, et al., presented a new model, named Active Appearance Model (AAM) [12-14], which overcomes the disadvantages of ASM by integrating complementary texture information from the image. In [13], the authors give a comprehensive review of the improvement of AAM from three different perspectives: effectiveness, discrimination and robustness. For the purpose of applying AAM in real-time situation, researchers developed many methods to reduce the computational complexities by involving texture representation, optimization algorithm and model

training. As a result, AAM brings considerable advances to statistical-model with new features of synthesizing picturistic images. AAM finds its wide applications in various situations, e.g. facial recognition and expression [15], human eye movement [16], object tracking[17], and medical image segmentation [74].

(a)

(b)

(c)

(d)
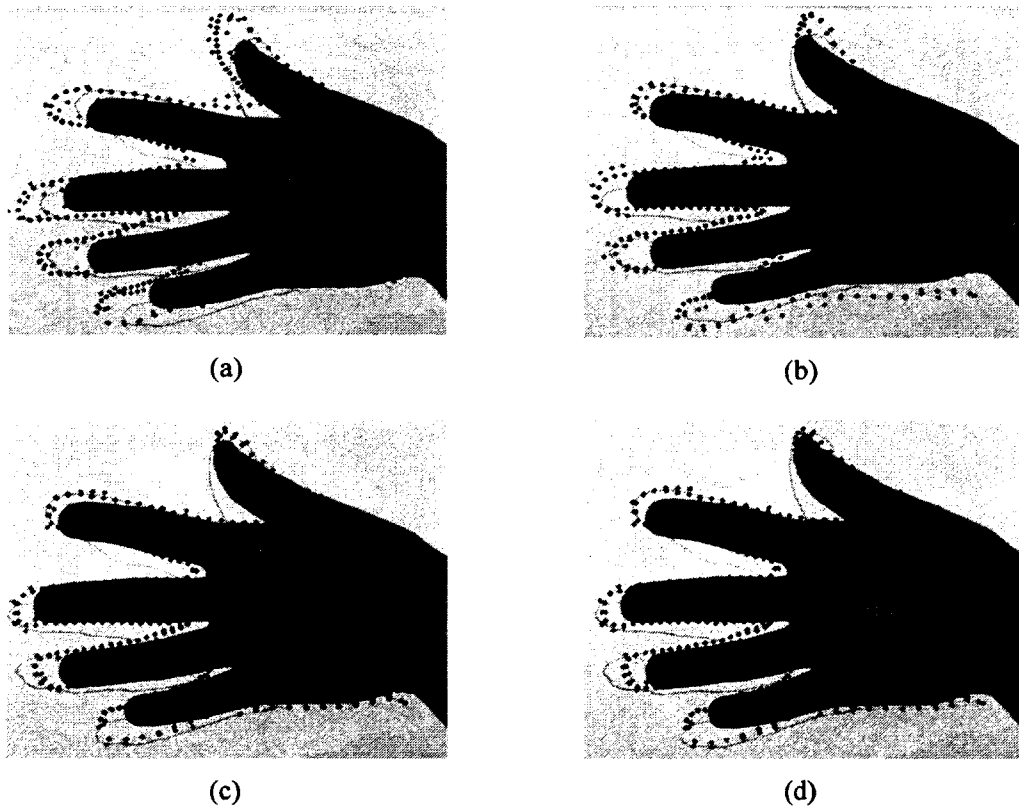
Fig. 3.1 Illustration of ASM applied to hand matching, (a) initialization,
(b)-(c) transitional results, (d) the final result

In [19], the authors point out that one of the drawbacks of AAM is that the pose modeling is the most difficult problem for shape modeling. The reason for this stems from the pose changes in 3D space, which lead to the nonlinear solution of facial appearance. The

authors also mention that the combination of 2D and 3D might be able to find the solution between computational complexity and effectiveness for AAM.

Another remarkable paper was published by Volker Blanz and Thomas Vetter in 1999 [6], in which the author presented a statistical-based method in the 3D space, which is named 3D Morphable Model (3DMM). The estimation process of fitting shape and texture information into 2D images is considered as the problem of searching for the global minimum in a high dimensional feature space, in which optimization is apt to have local convergence.

3.2 Introduction to Statistical 3D Morphable Model

3D face modeling has been one of the greatest challenges for researchers in computer graphics for many years. Various methods have been used to model the shape and texture of faces under varying illumination and pose conditions from a single given image. In order for better understanding of 3D Morphable Model, I would like to give a rough introduction of the previous works done in this field in the following several pages.

3.2.1 Previous Works

Researchers introduced Shape-From-Shade (SFS) [18] to reconstruct the 3D surface of human faces. Unfortunately, SFS shows rapid decrement in performance, such as biased calculations and improper estimates of surface normals caused by varied lighting conditions and cast shadows on the 2D image. The active appearance model (AAM), which has been mentioned in the previous section, is also a statistical deformable technique which has been widely used in computer vision.

(a)   (b)
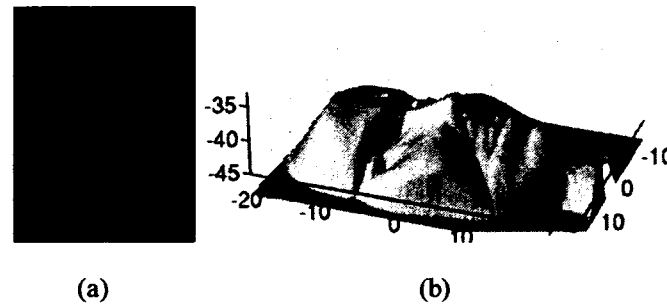
Fig. 3.2. Illustration of Shape-from-Shade model, (a) original image, (b) 3D shape with respect to (a)

However, AAM only allows a small range of out-of-plane rotation and displays inadaptation to directed light sources. In 1995, David Beymer and Tomaso Poggio created a pose-invariant face recognizer whose strategy is based on a set of example views at different poses. By using 2D example views of prototype faces under different rotations, they proved that it is still possible to use the view-based approach by exploiting prior knowledge of faces to generate virtual views, or views of the face as seen from different poses. During the process of deformation, they measured the prototype optical flow between face images of different views from one person and then mapped this prototype flow to the new person to generate the virtual view of the that person.

In 1999, Georghiades, et al. [4] proposed an image synthesis method based on illumination-cone to deal with pose and illumination problems. They build the concept that is able to solve the problem of illumination variations successfully in face reconstruction. But the issues, such as variations due to rotation, regarding pose variance still remain fully unsolved though they have suggested many ways to the problem solving.

In [19], Jinho Lee and *et al* obtain arbitrary human faces using a sequence of silhouette images as the input. They demonstrated that 2D silhouette matching captured in different viewpoints play important role in human face reconstruction. However, the performance of their experimental results is highly affected by the number and locations of cameras for the high qualified shape recovery. Romdhani and Vetter [20] point out in their paper that, in some cases, the intensity information by itself cannot prevent fitting process from stepping into local minima. Thus they include 3D Morphable Model extra features extracted from the image, like edge information and highlight specular, to constrain the optimization process more likely on the path toward the optimal result.

In [21], A. Patel and W. A. P. Smith addressed the combination of statistical tools of thin-plate splines and Procrustes analysis to construct a morphable model for face shape recovery.

In [22], the authors designed their method for using a single 3D reference model of a different person's face. Different from classical reconstruction methods from single images, which require knowledge of the reflectance properties and lighting as well as depth values for boundary conditions, in the proposed method a single reference model is molded by the input image reach a reconstruction of the sought 3D shape. Their method uses harmonic representations of lighting based on the assumption of Lambertian reflectance.

3.2.2 Face Representation and Synthesis

"A virtual view is a synthetically generated image of a face with a novel pose or

expression." [23] In 1992, Poggio and Vetter first proposed the idea of "linear class." Associated with the image representation that is based on pixelwise correspondence, they addressed a novel model for object representation, which uses linear combinations of views to define and model classes of objects. They were inspired individually by the results of Ullman and Basri [24] and Shashua [25] who both demonstrated that any virtual view of an object is able to be obtained by linear combinations of three views of that object.

In their paper, Poggio and Vetter defined a linear object class as a set of 2D views of objects that "cluster in a small linear subspace of R^{2n} where n is the number of feature points on each object." At the same time, they declare that in the case of linear object classes rigid transformations can be learned exactly from a small set of examples. In addition, other object transformations (such as the changing expression of a face) can be approximated by linear transformations in the same way. In particular, they used their linear model to generate new virtual views of an object from a single (example) view.

Vetter and Poggio combined a linear model of shape and texture vectors to generate virtual views across large viewpoint changes [26] In 1995, Beymer addressed a new approach for modeling human faces in order to generate "virtual" faces in a face recognition system [23]. In 1996, Ezzat made use of the linear combinations of vectorized images to build a model for synthesis and analysis of novel views of a specific face [27]. In the following section, we will review a few models based on the concept of *"linear class"* to represent a novel view of the object, particularly faces, by the synthesis a set of examplar images of that object from different views.

3DMM can not only be applied to face reconstruction from images but also from video frames. In [28], 3DMM is used for 3D face reconstruction from videos. Their method is basically based on the combination of Support Vector Machines (SVMs) and a Morphable Model of 3D faces. The algorithm first detects individual facial features by using a novel regression- and classification-based approach, and a list of candidates for several facial feature positions are selected from probabilistically plausible configurations of features. Then the configurations of feature points are evaluated using a novel criterion that is based on a Morphable Model and combinations of linear projections. Finally, the feature points initialize a model-fitting procedure of the Morphable Model.

3.3 3D Morphable Model

3D face modeling has been one of the greatest challenges for researchers in computer graphics for many years. 3D Morphable Model (3DMM), proposed by T. Vetter [29, 30], is a realistic 3D face modeling method which can be synthesized automatically by a linear combination of exemplar faces. One of the applications of 3DMM in face recognition is to create the 3D face model of an individual from given 2D images [30, 31].

The reconstruction procedure is regarded as conducting iterations of the analysis-by-synthesis process, which are driven by fitting the 3D model to 2D images. Meanwhile, the parameters with respect to 3D environment such as focal length of the camera, illumination and color contrast, can also be modeled explicitly and estimated automatically. As valuable as it is, morphable model, therefore, could be utilized to expand the spectrum and create versatile variations for the original photographic database.

3.3.1 Model Construction

The prototypical 3D faces are acquired by 3D laser scanners, whose range and texture data that are digitalized with high precision. Preprocessed through registration and texture extraction, each face is represented in the form of a shape vector and a texture vector as:

$$S = (X_1, Y_1, Z_1, X_2, ..., Y_n, Z_n)^T \tag{3.1}$$

$$T = (R_1, G_1, B_1, R_2, ..., G_n, B_n)^T \tag{3.2}$$

where n is the number of vertices on the 3D face and $(B_j \, G_j \, R_j)$ are the corresponding R, G, B color values of the vertex $(X_j \, Y_j \, Z_j)$. Therefore, a morphable model can be generated by using the linear combination of shape vectors S_i and texture vectors T_i of 3D training faces as [29]:

$$S_{model} = \sum_{i=1}^{m} a_i S_i \qquad T_{model} = \sum_{i=1}^{m} b_i T_i \qquad \sum_i^m a_i = \sum_i^m b_i = 1 \tag{3.3}$$

in which m is the number of training faces, S_i and T_i are shape and texture of training faces and a_i and b_i are their corresponding weights contributed to the new face with $0 < a$, $b < 1$.

$$S_{model} = \alpha_1 \cdot \boxed{} + \alpha_2 \cdot \boxed{} + \alpha_3 \cdot \boxed{} + \alpha_4 \cdot \boxed{} + \dots = S \cdot \alpha$$

Arbitrary face shape formation

$$T_{model} = \beta_1 \cdot \boxed{} + \beta_2 \cdot \boxed{} + \beta_3 \cdot \boxed{} + \beta_4 \cdot \boxed{} + \dots = T \cdot \beta$$

Arbitrary face texture formation

Fig. 3.3 Illustration of the formation of linear combination of 3D Morphable Model

In the practical consideration of computational effectiveness, a common technique as PCA (Principal Component Analysis) is employed to reduce the high dimensionality of 3D face data without the loss of potential face information. In particular, PCA performs a transformation of the original cloud data to an orthogonal coordinate system formed by the eigenvectors s_i and t_i of the covariance matrices.

$$\mathbf{S}_{model} = S_{mean} + \sum_{i=1}^{m-1} \alpha_i \mathbf{s}_i \qquad\qquad T_{model} = T_{mean} + \sum_{i=1}^{m-1} \beta_i T_i \qquad (3.4)$$

where S_{mean} and T_{mean} are the average shape and texture vectors. S_i and T_i are principal components. $\alpha = (\alpha 1, \alpha 2, ..., \alpha n)$ and $\beta = (\beta 1, \beta 2, ..., \beta n)$ are shape and texture combination coefficients, and α and β obeys Gaussian distribution as:

$$p(\alpha) = \exp\left(-\frac{1}{2}\sum_{i=1}^{m-1}\left(\frac{\alpha_i}{\sigma_{S,i}}\right)^2\right) \qquad\qquad p(\beta) = \exp\left(-\frac{1}{2}\sum_{i=1}^{m-1}\left(\frac{\beta_i}{\sigma_{T,i}}\right)^2\right) \qquad (3.5)$$
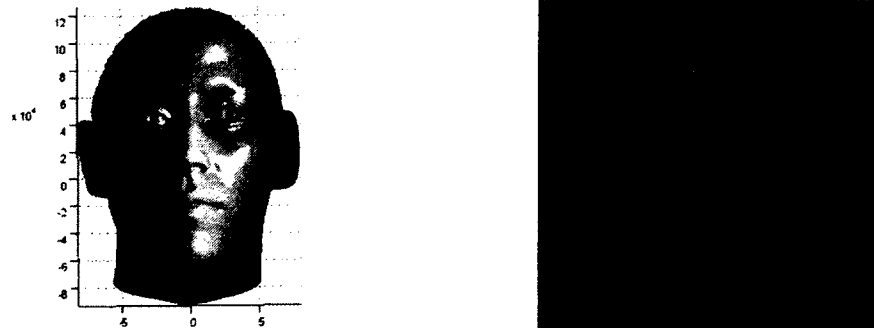
Fig. 3.4 Shape and texture information of the training face
(a) shape information (b) the corresponding training face.

3.3.2 Several Key Points in 3DMM

3DMM is a composite and sophisticated model that consists of a number of various techniques in computer vision and computer graphics. To make a clear concept of 3DMM, we would like to discuss several key factors of the model as well as the associated considerations in our implementation.

Similar to AAM, 3DMM decouples the shape and texture information and then analyzes them separately. PCA is the most common way in statistical-based method for model analysis, which is inherited by 3DMM with extension to the 3D space. Since PCA has the ability to capture the most significant features of the training data and convert them into the PC space or the feature space, it is possible to reduce the dimensionality of 3D data into the feature space with lower dimensions. Even more, it ensures concentration on the most significant features without minimum loss in the result quality. We emphasize this point here, because we take advantage of this property in our experiments, and which results in the fact that only the first 70s features from the principal components of shape and textures are included. In Figure 3.5, we display a sequence of projections of a synthesized 3D face model with respect to different amount of principal components of shape and texture involved simultaneously. The images illustrate that with fewer PCs added to the model, the face is located closer to the mean face; whereas when more PCs are taken into consideration, more and more details show up in the model.
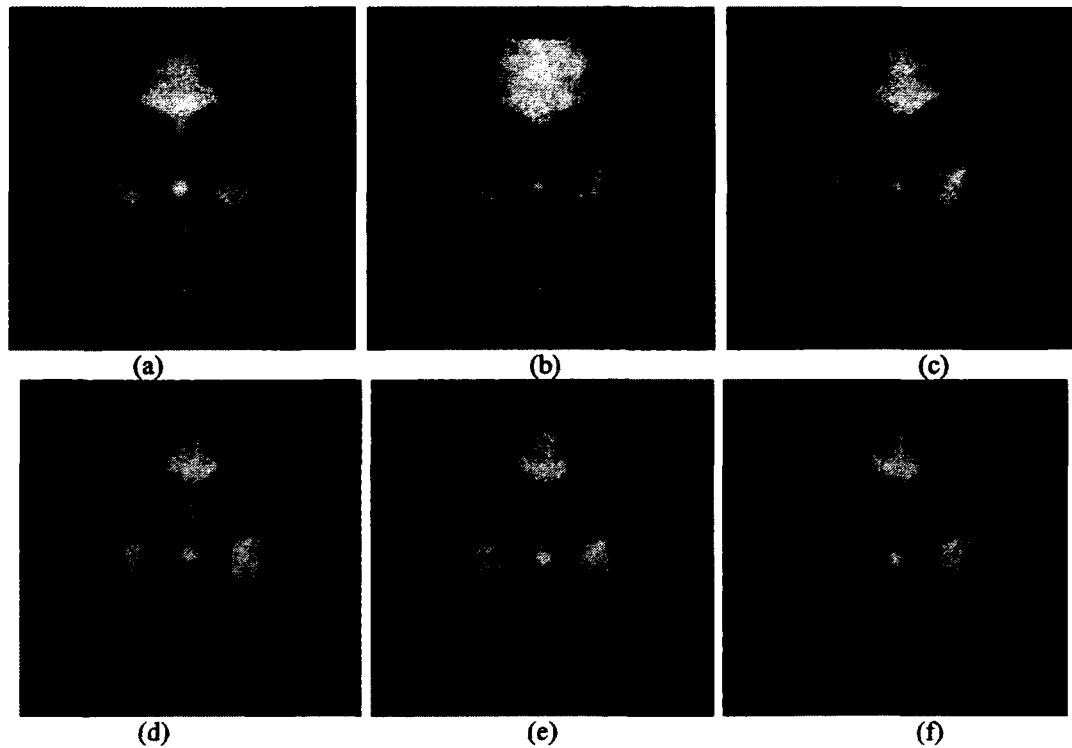
Fig. 3.5 (a) face model generated by different PCs,
(a)-(f) face model with 5,10,20,30,40,50 PCs respectively

Another key point that is necessary to stress here, is the importance of generating correspondences between the 3D face model and the target image. Based on the observation from our experiments, the correspondence issue, to some extent, is a crucial factor to the final result of the whole system. It has been proved by our experimental results that the 3D face model and the target image are supposed to be well registered before the fitting procedure starts; otherwise there is little chance to achieve sound performances. In the next section, we will discuss more details about our findings on this. At the same time, we will present our own approaches to the issue of the optimal correspondence.
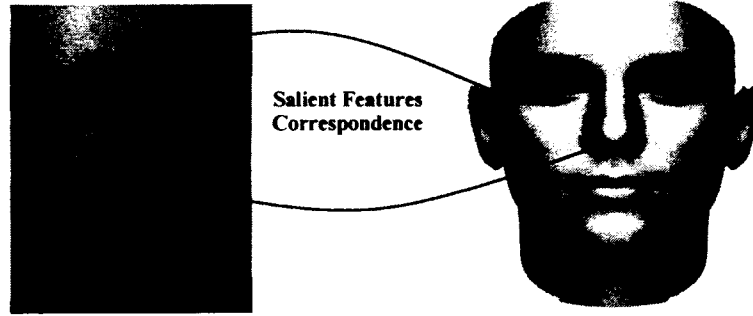
Fig. 3.6 Feature points correspondence
between 3D model and 2D target image

Thirdly, as the face model is manipulated in the context of 3D space, it is inevitable to deal with the illumination models or lighting models. In order to create a realistic 2D output that is close enough to the target face image, we make use of the perspective projection and Phong illumination model in the rendering process. Given the kth vertex at (X,Y,Z) with texture value (R,G,B), the perspective projection on the image plane is represented as:

$$I_k(x,y) = \left(I_{r,k}(x,y), I_{g,k}(x,y), I_{b,k}(x,y) \right)^T \qquad (3.6)$$

where $I_{c,k}(x,y)$ is computed under the Phong illumination model as:

$$I_{c,k}(x,y) = R\left(I_{a,c} + I_{dir,c}(L \cdot N) \right) + K_s\, I_{dir,c}(F \cdot V)^n \qquad (3.7)$$

$I_{a,c}$ and $I_{dir,c}$ are separately intensity of ambient light and direct light of the cth color component. K_s is the reflectance, L, N, F, V are light direction, normal, reflective direction and direction of viewer respectively and n is the mirror reflectance index.

The other problem should be taken into consideration is how to determine the mode of 3D projection that maps 3D objects to the 2D image plane. Orthographic projection is the

one where a 3D object is projected onto a plane by a set of parallel rays orthogonal to the

image plane. Equation (3.8) is used to represent the orthographic projection.

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(3.8)

in which $x = \dfrac{x_h}{w} = X$ and $y = \dfrac{y_h}{w} = Y$.

Different from the orthographic projection, the perspective projection brings the effect

that objects in the distant appear smaller than objects close by. Equation (3.9) is used to

represent the perspective projection.

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

(3.9)

in which $x = \dfrac{x_h}{w} = \dfrac{fX}{Z}$, $y = \dfrac{y_h}{w} = \dfrac{fY}{Z}$ and $z = \dfrac{z_h}{w} = f$.

However, the classic perspective projection mentioned above is too complicated to

implement. So we take an alternative projection mode instead, called weak perspective

projection that is a simplified equation of the classic perspective projection. The weak

perspective projection assumes that the object's dimensions are much smaller to its

average distance Z_{avg} to the camera. As a result, x_h and y_h are proportional to the distance

from the object to the camera. Equation (3.10) is used to represent the orthographic projection.

$$
\begin{bmatrix} x_h \\ y_h \\ z_h \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Z_{avg} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}
\tag{3.10}
$$

in which $x = \dfrac{x_h}{w} = \dfrac{fX}{Z_{avg}}$ and $y = \dfrac{y_h}{w} = \dfrac{fY}{Z_{avg}}$.

CHAPTER 4

GLOBAL OPTIMIZATION FOR 3DMM

4.1 Introduction

The morphable model provides an approach in solving face modeling problems under different illumination and pose conditions. However, one of the issues lies in the process of minimizing the cost function that performs error evaluation in the pixel-level measurement. This involves algorithms of image matching and a large-scale optimization. In 3DMM, fitting shape and texture into 2D images is equal to searching for the global minimum in a high dimensional feature space, in which optimization is apt to have local convergence. Stochastic gradient descent [29] and Levenberg-Marquardt [20] method are used to evaluate the residual and global error as well as objective function optimization. Differential Evaluation (DE) appears to be robust against stagnation in local minima and sensitiveness to initial values in face reconstruction. Considering its successful performance, we tentatively introduce DE to tackle the problem in 3D-2D matching.
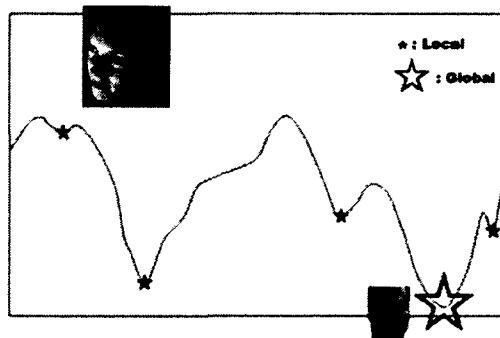

Fig. 4.1 Illustration to the global optimization for 3D-2D Fitting, the image on the top left is denoted as the initial state of the mean face; the image on the right bottom is the target face; the hollow star symbol indicates the global minimum while the solid star indicates the local minimum

4.2 Global Optimization

Global Optimization is a set of approaches whose main task is to determine the optimal global solution to the objective function f, which seeks its optimum under the condition of a set of equality and inequality constraints. Unlike finding the solutions that are locally optimal, global optimization is considerably more challenging to evolve into the vicinity of the global convergence point. For all that, in the past decade, global optimization has found a number of ever-growing facets of application, such as engineering, applied sciences, and sciences [32] [56]. Global optimization algorithm could be divided into a number of categories according to their method of operations [64], such as deterministic algorithms and probabilistic algorithms. In the latter scheme, Monte Carlo algorithms play very important roles, which contains one of the most important classes of global optimization, Evolutionary Computation.

4.2.1 Evolutionary Algorithms

Evolutionary Algorithms (EA) is a population based global optimization scheme inspired by the biological mechanisms and processes, e.g. mutation, crossover and selection. EA contains a big set of members, such as Genetic Algorithms (GAs), Genetic Programming (GP), Learning Classifier System (LCS), Evolutionary Programming (EP) and etc, which have following features in common [64]:

1. The population size, denoted as NP, or the number of populations used.

2. The method of selecting the individuals for reproduction, denoted as f.

3. The way the offspring is included into the population(s).

4.3 Differential Evolution

Differential Evolution (DE) is a "parallel direct search method" [33], which was first proposed by Storn and Price in 1995. It is characterized as a stochastic and population-based global optimization that is simple and effective for implementation. It inherits the merits of employing the difference between vectors in the objective function from two ancestors, the Nelder-Mead algorithm [34, 35], and the controlled random search (CRS) [36]. Unlike general Evolutionary Algorithms (EA), DE applies variations (differentiation and crossover) sequentially to each individual whilst EA is more apt to stand a better chance to reproduce the parent generation itself.

4.3.1 Background and Previous Works

For better clarification of DE, we list several principal advantages of DE over other global optimization algorithms.

a. In contrast to other EAs, DE demonstrates its advantages of straightforward for implementation, whose simplicity arises from a concise computational framework of iterations. In other words, it reduces the threshold and the associated workloads of users for coding. Figure 4.2 depicts the operational procedure of standard DE.

b. The successful performance of DE makes it stand out from other optimization algorithms. It has been proved in studies [37, 38] that DE outperforms other algorithms, like G3 with PCX, MA-S2 and etc. According to the report from, DE was proved to be the fastest evolutionary algorithm in the competition. Additionally, in a comparison by Storn and price in 1997, DE surpasses simulated annealing and genetic algorithms by more efficiency. In 2004, Ali and Torn revealed that DE was both more

accurate and more efficient than controlled random search and another genetic algorithm. In the same year, Lampinen and Storn demonstrated that DE was more accurate than several other optimization methods including genetic algorithms, simulated annealing and evolutionary programming. Considering the overall situations, DE is significantly impressive in terms of accuracy, converged speed, reliability and etc.

c. Global optimization is, in practice, generally taken into consideration in the scenario of finding optimal solution to high dimensional vectors, in which easiness of manipulation is concerned as one of the major factors. The whole framework of DE is controlled by few parameters, such as F, CR and NP. Even for its varieties developed in later years to increase the DE's performance, the total *overhead* applied to the control parameters is relatively less than other global optimization models.

From the viewpoint of algorithm, the secret of DE's success was revealed by [39], in which the author summarizes it as *spontaneous self-adaptability, diversity control and continuous improvement.*

DE discovers itself wide ranges of applications in various areas. For example, DE was applied to the multisensory fusion problem [40], the power electronics problem [41], data mining [42]. In addition, it has been utilized to DNA researches to discover subsets of informative genes that accurately characterize all the samples [43]. It is also applicable to the integration of other algorithms, such as digital filter design [44], neural network training [45] and so on. A review of DE applications is presented in [46].

4.3.2 Introduction to DE Algorithm

DE repeatedly processes through operations which are, in turn, "mutation, crossover and selection" until an optimal solution to the objective function $f(x)$ is reached (Figure 4.1).
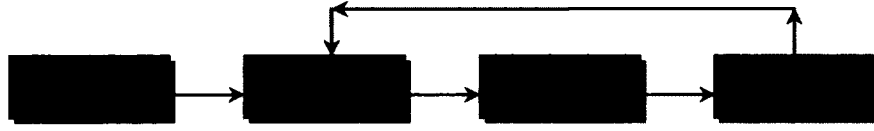
Fig. 4.2 Canonical Differential Evolution Procedure

The classic version of DE is defined as follows. Suppose we have N D-dimensional parameter vectors

$$x_{i,G} = \left[x_1, x_2, ..., x_D \right]^T, i = 1, 2, ..., N \tag{4.1}$$

representing the population for generation of G. The algorithm starts by randomly initializing the vector populations with, as the author suggested, a uniform probability distribution [47]. We use a different distribution in our experiment due to the special feature of 3DMM, which we will present later in Chapter 4.

Step 1: Initialization

The vector population is randomly initialized with uniform probability distributions.
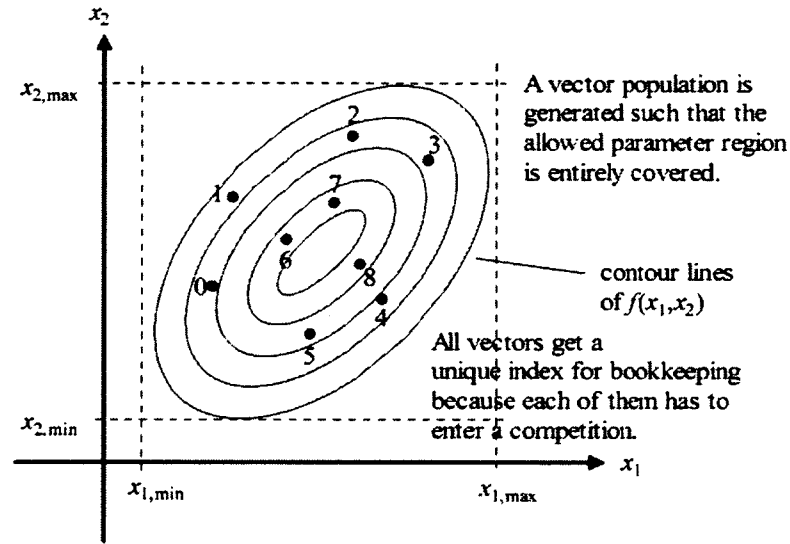
Fig. 4.3 DE population initialization
(Courtesy of Differential Evolution - A Practical Approach to Global Optimization)

Step 2: Mutation

For each individual x_i, a corresponding mutation vector v_i is produced according to the equation:

$$v_{i,G} = x_{r1,G} + F * \left(x_{r2,G} - x_{r3,G} \right) \tag{4.2}$$

in which random index $r1, r2, r3 \in \{1, 2, ..., N\}$ and $r1 \neq r2 \neq r3$. F is a real amplifier designed to control the offset of $v_{i,G}$ to $x_{r1,G}$ by scaling the differential variation $\left(x_{r2,G} - x_{r3,G} \right)$.

There is another mutation strategy developed by Price in 1996, where

$$v_{i,G} = x_{best,G} + F * (x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G}) \tag{4.3}$$

The involvement of another differential vector is to expand the range of optimization searching in the case of population NP that are high enough.
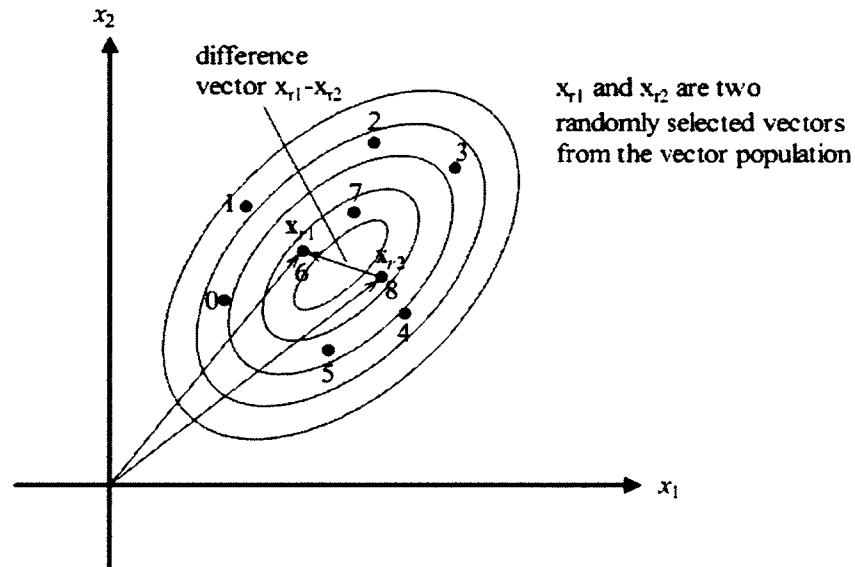
Fig. 4.4 Generate the perturbation: $x_{r1} - x_{r2}$
(Courtesy of Differential Evolution - A Practical Approach to Global Optimization)
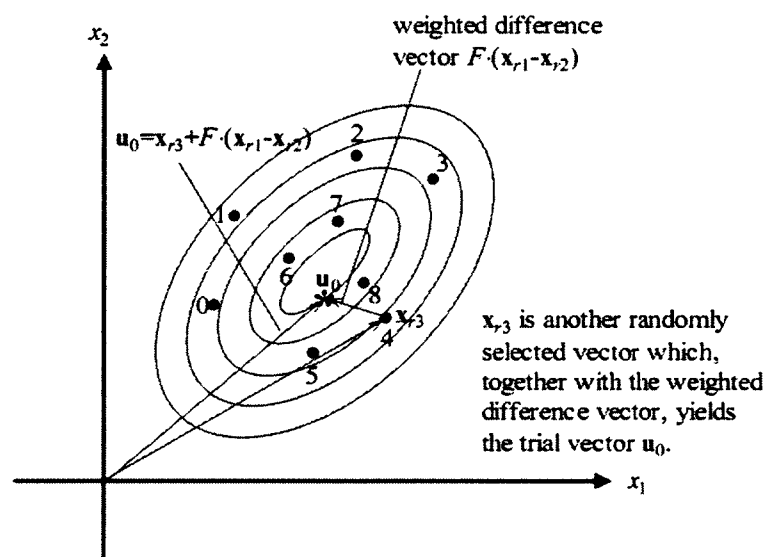
Fig. 4.5 Mutation
(Courtesy of Differential Evolution - A Practical Approach to Global Optimization)

Step 3: Crossover

Trial vectors are introduced in the phase of crossover to expand the range of global search. It is defined in the form as:

$$u_{i,G} = \left(u_{1i,G}, u_{2i,G}, \ldots, u_{Di,G} \right) \qquad (4.4)$$

in which

$$u_{ji,G} = \begin{cases} v_{ji,G}, & if\left(randb(j) \leq CR \right) \vee j = I_{br} \\ x_{ji,G}, & if\left(randb(j) > CR \right) \wedge j \neq I_{br} \end{cases} \qquad j = 1,2,\ldots,D. \qquad (4.5)$$

In equation (4.5), $randb(\cdot)$ is a random generator with uniform distribution. CR is the user

defined crossover constant $\in [0, 1]$. I_{br} is an integer randomly chosen from $\{1,2,\ldots,D\}2$,

which prevents $x_{i,G}$ from being equal to $u_{i,G}$.

Step 4: Selection

DE utilizes pair-wise comparison between $u_{i,G}$ and $x_{i,G}$ to survive the vectors with fewer

objective function values to the next generation.

$$x_{i,G+1} = \begin{cases} u_{i,G}, & if\left(f\left(u_{i,G}\right) < f\left(x_{i,G}\right) \right) \\ x_{i,G}, & otherwise \end{cases} \qquad i = 1,2,\ldots,N \qquad (4.6)$$
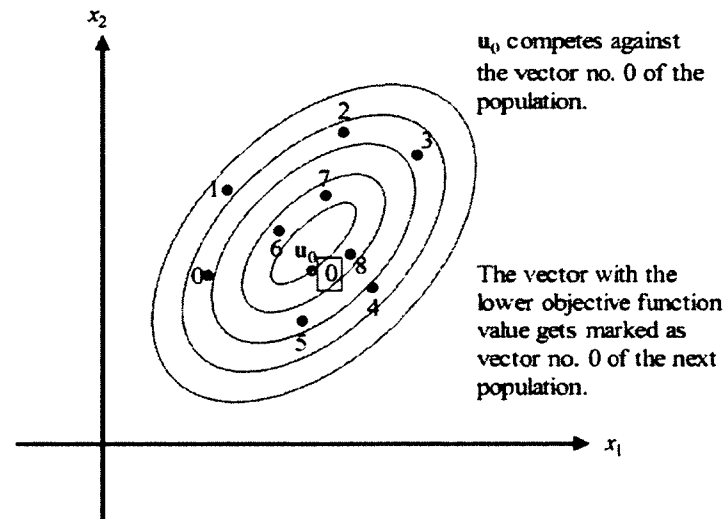
Fig. 4.6 Selection. u_0 replaces the vector with index 0 in the next generation since its lower function value. (Courtesy of Differential Evolution - A Practical Approach to Global Optimization)
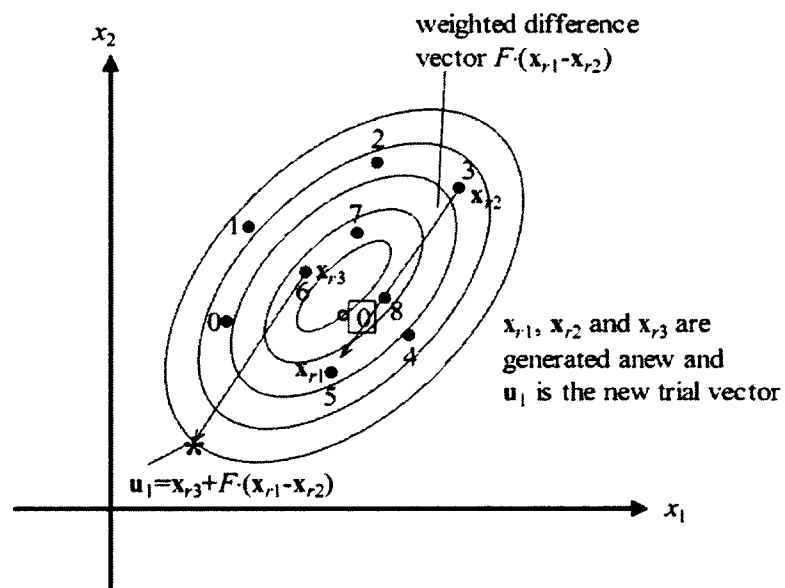
Fig. 4.7 A new population vector is mutated with a randomly generated perturbation. (Courtesy of Differential Evolution - A Practical Approach to Global Optimization)
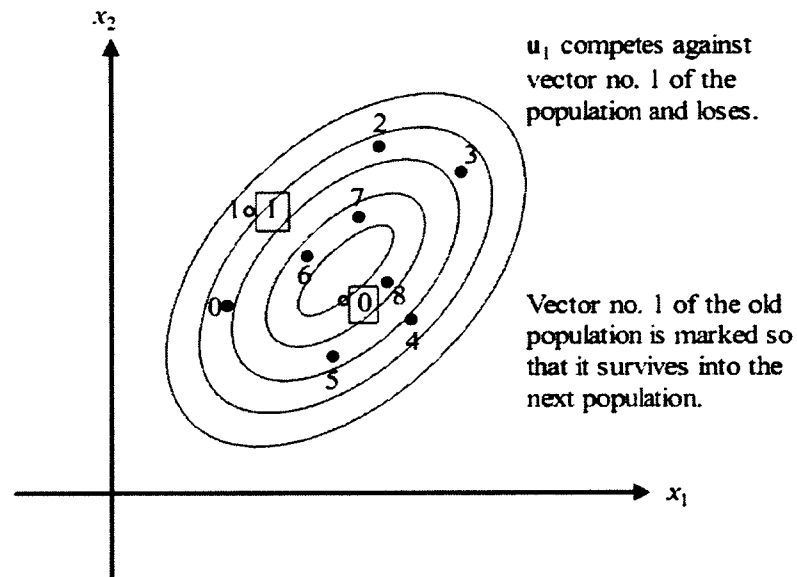
Fig. 4.8 Selection with the trial vector loses.
(Courtesy of Differential Evolution - A Practical Approach to Global Optimization)

In the following sequence of images (Figure 4.9), we can have a clear and straightforward explanation of the scheme of DE and how it works on finding the global minimum to the peak function.
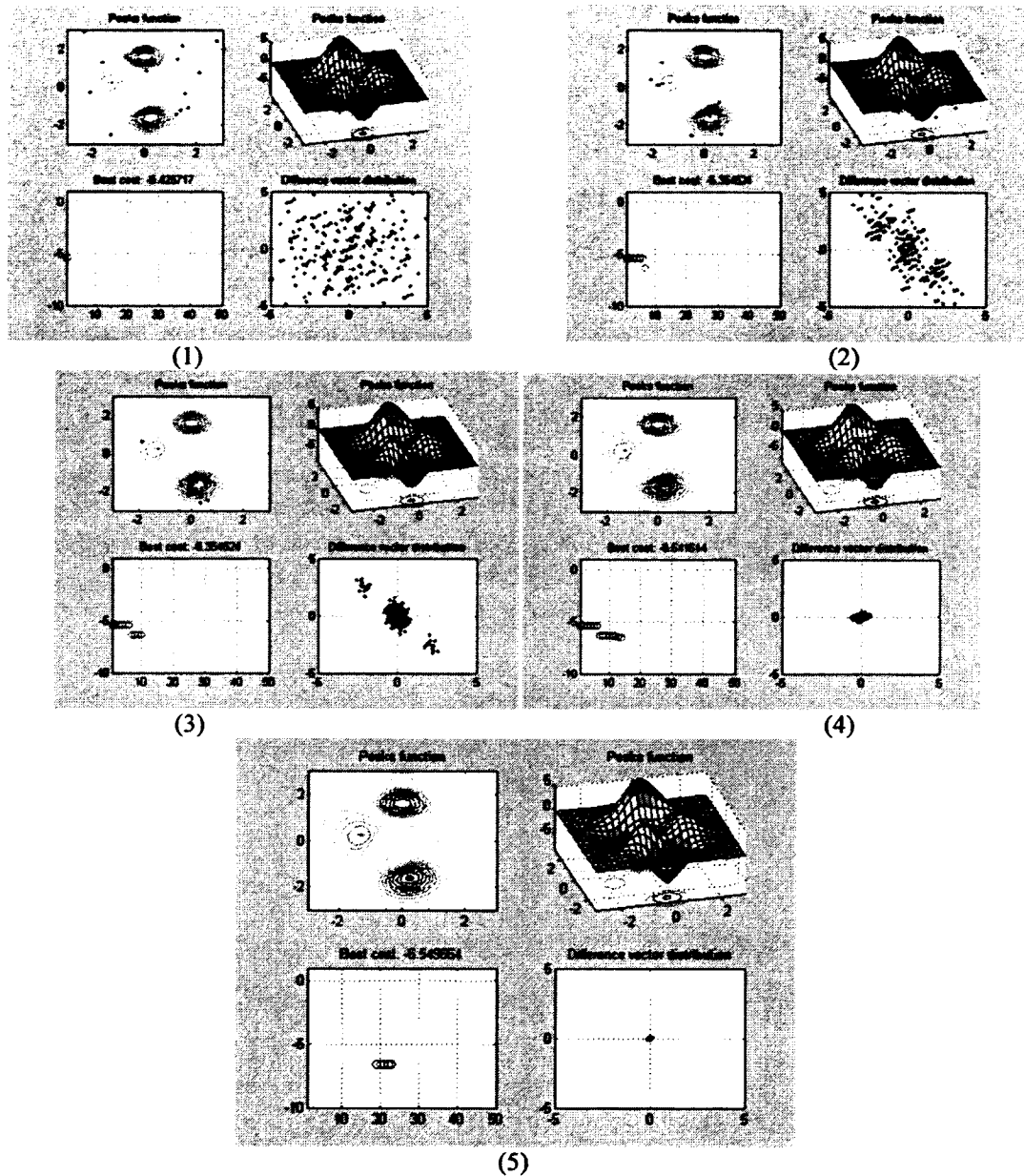
Fig. 4.9 DE optimization of finding global minimum of the peak function.
NP = 15, F = 0.8, CR = 1

4.3.3 Other Types of DE Algorithms

The classic DE algorithm is improved by many mutations for real world applications. The improvements are roughly categorized into several aspects, such as variables *NP*, *F* and *CR*, perturbation, diversity enhancement, vector populations and etc[46].

The choice of the variables of *NP*, *F* and *CR* is very important to the final result of the global optimization because it is highly related to the convergence, searching ranges and the time consumption. Although Storn and Price have pointed out the basic range of these parameters that would be suitable for most of the cases, there are still reports about finding the best *NP*, *F* and *CR*. In [23], the search range given by Storn and Price is (1) $F \in$ [0.5, 1.0], (2) $CR \in$ [0.8, 1.0] and (3) $NP \in$ 10*D. It is also proven in our experiments of 3D face reconstruction that the parameters in this range are more likely to find the best solution compared to other choices. In [15], an important criteria developed by the author is the mutation scale factor *F* should not less than F_{crit}, where

$$F_{crit} = \sqrt{\frac{1 - CR/2}{NP}}$$

(4.6)

Another very important factor of mutation scheme is the perturbation factor in the process of mutation. It is noticed equation (4.7) and equation (4.8) are commonly used in many applications of DE. Comparing to equation (4.7), equation (4.8) is *greedier* in the global searching.

$$v_{i,G} = x_{best,G} + F * \left(x_{r2,G} - x_{r3,G} \right)$$

(4.7)

$$v_{i,G} = x_{i,G} + F * \left(x_{best,G} - x_{i,G} \right) + F * \left(x_{r2,G} - x_{r3,G} \right)$$

(4.8)

CHAPTER 5

3D FACE RECONSTRUCTION

5.1 Key Factors in 3D Reconstruction

5.1.1 Face Detection

Prior to carrying on the 3D face reconstruction, it is of foremost importance to extract the region containing face information of the incoming image from its background. The major task is to determine the likelihood and the location of the face in the image before it is conducted to the procedure of the image analysis and understanding. Due to its importance to the automation of face reconstruction, we would like to introduce the state-of-art technologies of face detection as well as the one applied in our framework.

The face detection technology discovers its difficulties in dealing with the mutable image context, such as face variations in scale, location and orientation, occlusion and so on [48]. It has been pointed out that the research of face detection in the circumstances of unconstrained settings is confronted with challenges from many factors, especially the varied significance in pose and lighting. Fortunately, many important algorithms have been invented to cope with these uncertainties. One of the most famous and popular used algorithms is the Viola-Jones face detector, which is renowned for its real-time computing, simple implementation and sound accuracy. The algorithm achieves its successfulness based on three steps, e.g. the integral image, AdaBoost classifier and the cascade structure. Support Vector Machine (SVM) has also been introduced into the face detection as its excellent performance in seeking for the minimized empirical error and

the maximized geometric margin [24] [49]. The Face detector based on neural networks is one of the alternatives which are trained to build up a generic model by reducing the dimensionality of the features of facial representation [50].

In our framework, we utilize the Viola-Jones face detector to find the face region in the incoming image. In addition, we are able to find the region of eyes and mouth, which provides the possibility of locating the feature points. The region, marked by the rectangle box in Figure 5.1, is the final results from Viola-Jones face detectors. The detailed information about the Viola-Jones face detector is introduced in the Appendix A.

Fig. 5.1 Illustration of face detection using Viola-Jones face detector

5.1.2 Feature Points Correspondence

As we discussed in Chapter 2, correspondence plays an important role in the 3D-2D matching, which is favorable to the 3D face finding an optimal initial point to begin with. Considering the instinctive property of human faces, we focus the correspondence by locating salient facial feature points on the 2D image and the 3D face model, which is able to represent most of the face attributes. As we can see from the illustrations below,

the feature points include eyes corner, nose tip, mouth corner and a few marks on the facial outline. For 2D face images, we manually locate annotation landmarks with the associated facial appearance. For the 3D face model, we manually select the corresponding vertices on the mean face model shown in Figure 5.2. We use mean face model for correspondence because it is the basic shape of the model morphing. On the other hand, mean face model represents as a rigid object which makes it easier to process for 3D transformation.

Fig. 5.2 Annotation landmarks manually selected
on the 2D image and 3D face model

Facial appearance on incoming images is fairly unpredictable. Orientation variations are the most common situation encountered. Hence, we impose a number of 3D rigid transformations, such as rotation, translation, scaling and projection, to cope with different poses of the facial appearance. For example, the angles of rotations along with different axis, R_α, R_β and R_γ; translation along x, y, z, T_x, T_y and T_z; scaling along x, y, z, S_x, S_y and S_z. Unfortunately, as perspective projection is a nonlinear computation, it can not be solved by linear equations. We follow the ideas of optimization and incorporate this issue into the DE optimizer. We design the cost function E_{marks} as the Euclidean

distance between the 2D landmarks and the projective position of the 3D landmarks on

the 2D plane, which is in the form as,

$$E_{marks} = \sum_{x,y} \left\| L_{img}(x,y) - proj\left(M_{3d} * L_{model}(x,y,z)\right) \right\|^2 \quad (5.1)$$

where M_{3d} is the sequential 3D transformations of rotation, translation and scaling applied

to the 3D model. $Proj(\cdot)$ is the perspective projection. $L_{img}(x, y)$ indicates a set of

locations of 2D landmarks in the image space.

As long as the optimal solution is achieved, the 3D model face is ready sitting on the right

position and facing the right orientation to start the appearance matching. Figure 5.3

depicts conceptually the optimization process of mean face orientation to the image facial

orientation. Please note that some feature points are ignored because of the facial self-

occlusions.

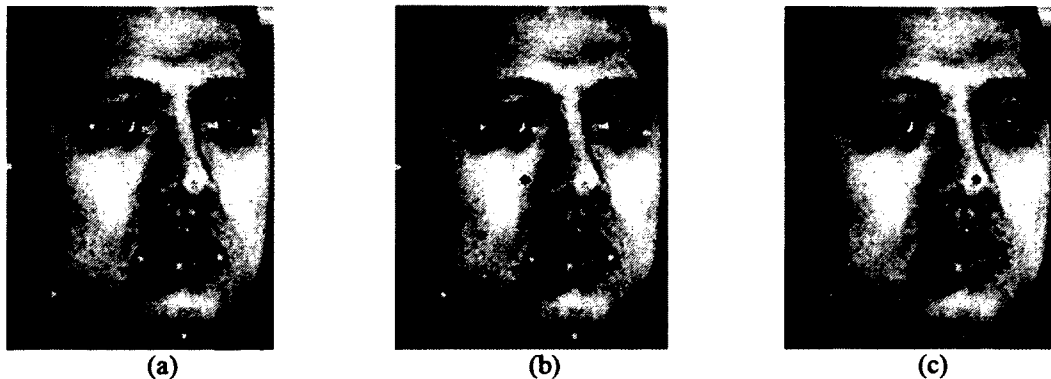(a)   (b)   (c)

Fig. 5.3 Result comparison before and after the shape registration, (a) the landmarks on the 2D image,
(b) overlaid landmarks of initial position of mean face model on (a)
(c) overlaid landmarks of the optimization result on (a)

Figure 5.4 shows the learning curve of the optimization of orientation estimation using

DE with population NP = 50, F = 0.8 and CR = 0.5.
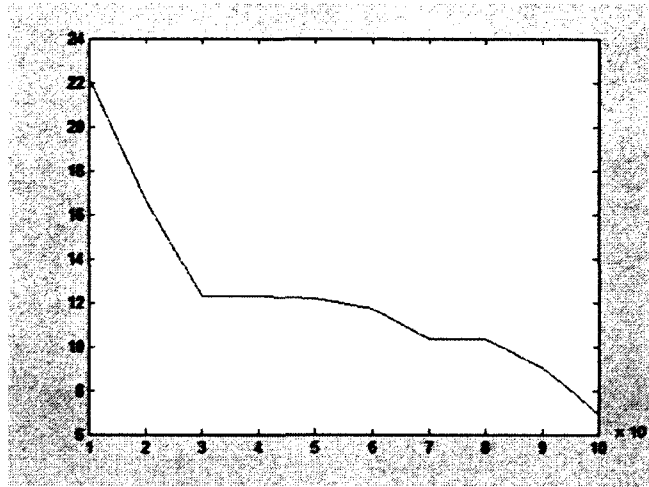
Fig. 5.4 Learning curve of the orientation estimation using DE

5.1.3 Color Fitting of 3DMM

The mean face of the 3D model (shown in Figure 5.2(b)) is created by averaging the prototypical 3D faces acquired by the 3D digital scanner. The color value of each vertex has already been determined at the time of acquisition, whereas the practical situation is the facial appearance of the image is diversified and cannot be decided beforehand. Inspired by the idea of histogram matching, we conclude a parametric way to fit the color of the model to the image.

The color images are commonly represented in the color space of $<R, G, B>$, in which images can merely be regarded as a set of pixel values. As a matter of fact, we can decompose an image in different color space to retrieve more knowledge hidden in the image. For example, we can convert the image from the $<R, G, B>$ space to $<H, S, I>$ space, in which H as *Hue*, S as *Saturation* and I as *Intensity*.

$$r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$$

$$h = \cos^{-1}\left\{ \frac{0.5*\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}} \right\}, \qquad h \in [0, \pi] \text{ for } b <= g$$

$$h = 2\pi - \cos^{-1}\left\{ \frac{0.5*\left[(r-g)+(r-b)\right]}{\left[(r-g)^2+(r-b)(g-b)\right]^{1/2}} \right\}, \qquad h \in [0, \pi] \text{ for } b > g$$

$$I = i*255; \qquad s \in [0, 1]$$

$$i = \frac{(R+G+B)}{(3*255)}; \qquad i \in [0, 1]$$

$$H = \frac{h*180}{\pi}; \quad S = s*100; \quad I = i*255$$

We use HSI color model because not only can H channel represent the global color tone of the image, but also I channel is able to provide edge information which normally contains promising features of the face image. How to integrate the estimation of model's color in our DE framework? We come up with an idea of using parametric equations to control each channel of H, S, I. Every channel is adjusted by two parameters as offset and amplifier, which is written as:

$$H = (H + offset_{hue})*Amp_{hue}$$

$$I = (I + offset_{int})*Amp_{int}$$

$$S = (S + offset_{sat})*Amp_{sat}$$

Then we integrate these six parameters in our DE optimization framework as the training

vector. Fig 5.5 depicts the channels of H, I, S respectively of the input image.
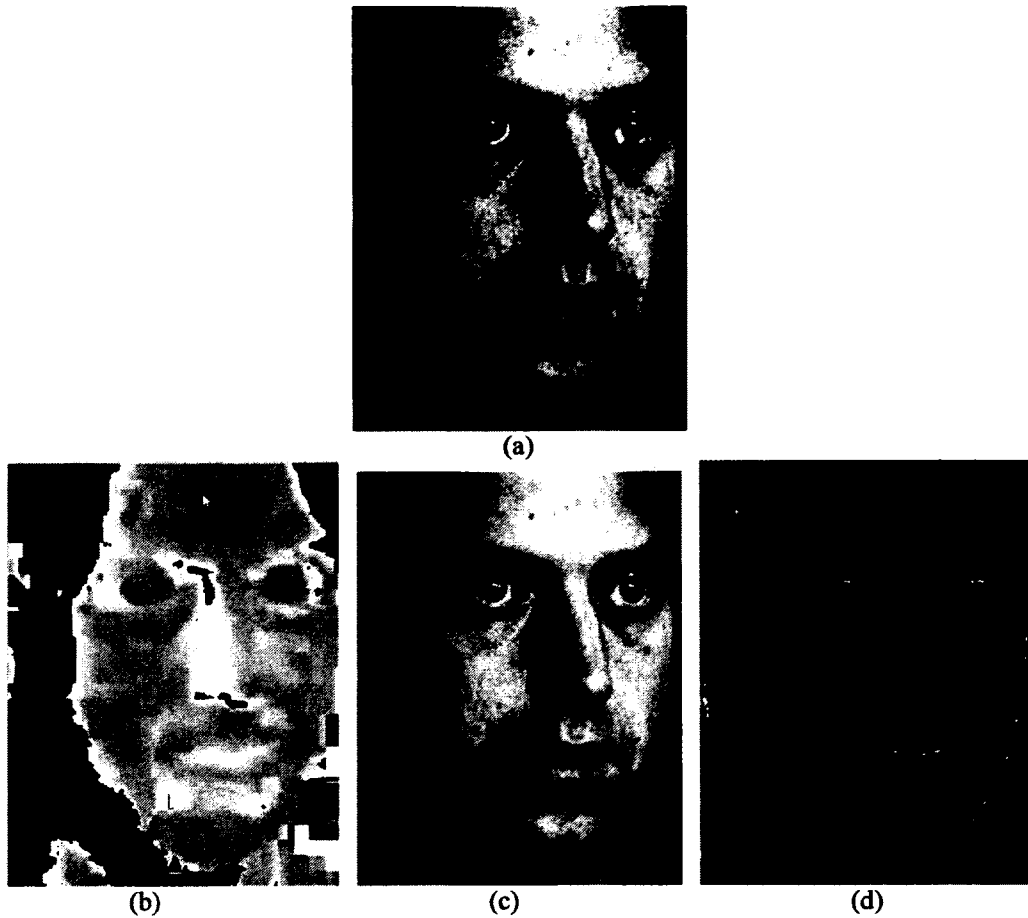
Fig. 5.5 Illustration of HSI transformation (a) the original image,
(b) – (d) hue, intensity, saturation respectively

5.1.4 Preprocessing of incoming images

Normally speaking, incoming images are coming with noise, which is an potential

troublemaker for the system to read useful information from the image. More importantly,

redundant noise highly increases the risk for the optimization to be divergent from the

optimal solution. So it is necessary to preprocess the incoming images for better

understanding of the image.

Regular lowpass filters can filter out noises but at the same time the distinctive features such as eyes edges or mouth edges will disappear from the image. In our work, we employ diffusion filtering to denoise the incoming image. Diffusion based image filtering methods [51, 52] make it convenient to screen out noises on the image, and simultaneously, it reserves, sometimes even enhances, the edge information. This merit is somewhat a complementary measure to contribute to the convergence of the optimization. Figure 5.6 makes comparison between images before and after the process of diffusion filtering as well as the image filtered by Gaussian filter, in which Figure 5.5(b) is obtained after 20 iterations of computation.

(a)   (b)   (c)
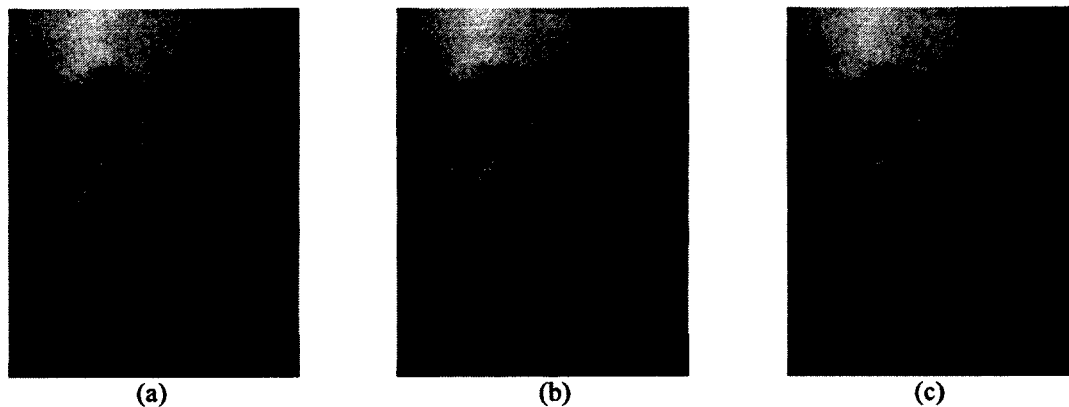
Fig. 5.6 Comparison between Diffusion filtering and Gaussian blur,
before(a) and after(b)the image Diffusion filtering, (c) after Gaussian blur

5.2 Experimental Results

Various 3D face databases have been established during the last decades. As far as we know, the current existing 3DMM databases are those from 3D Basel Face Model (BFM), Max Planck-Institute Tubingen (MPI) [53] and the University of South Florida (USF) . In our experiments, we use the 3D Basel Face Model (BFM) database [54] to derive the morphable model for 3D shape and texture. The reason we use BFM database is based on

three factors. Firstly, comparing to the other two, the BFM not only offers a higher resolution and precision than the MPI and USF models, but also makes use of a different registration method yielding less correspondence artifacts. Secondly, renderings of fitting results are more realistic with the new model. Thirdly, according to our experiments on the USF database, we found texture problems, which are unsuitable for reconstructing reasonable texture information.

The BFM database collects 200 3D faces from 100 male and 100 female subjects, each of which keeps neutral expression, without makeup, accessories and glasses. The 3D range data, which consists of shape and texture information, are collected through 3D scanners. For each vertex (x, y, z), there is a corresponding pixel value, denoted as (r, g, b). The 3D data is originally stored in the type of cylinder coordinate which brings more convenience to the accurate and detailed description of human faces. This feature also inspires the idea of converting the facial texture space to the image space, which we will talk about in Chapter 6.

The registered 3D faces are parameterized as triangular meshes with 53490 vertices [54]. Figure 5.7(a) shows the mean face of the 200 faces in the database, which is represented as S_{mean} and T_{mean} in (4). In equation (3.3), we have discussed the 3D faces are compliant with the manipulation of PC vectors, α and β. In Figure 5.7(b), we demonstrate 3D face synthesized by the randomly assignment of PC vectors.
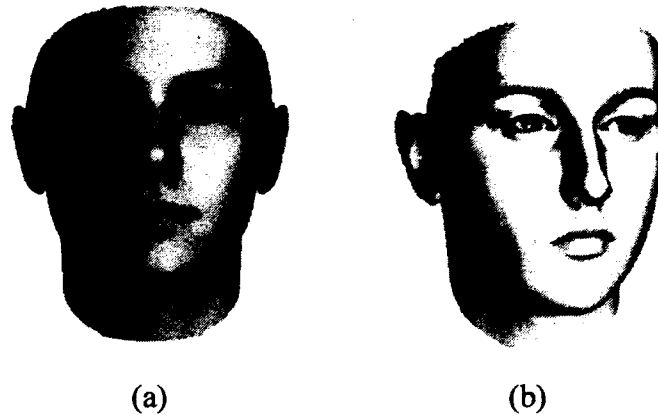
(a)   (b)

Fig. 5.7 (a) 3D mean face from BMF database, consists of 100 male faces and 100 female faces, (b) face generated by random assignment of α and β

5.2.1 3D Face Model Training

As described in Chapter 3, the 3D morphable model equipped with DE leads to our framework for 3D model reconstruction. In our experiment, DE populations are initialized by Gaussian distribution instead of uniform distribution. The reason for this is, according to the equation (3.5), shape and texture combination coefficients, α and β, both obey Gaussian distribution.

BFM database provides 200 principal components for both shape and texture representations. Due to the consideration of computational efficiency, we only employ the first 70 principal components of S_i and T_i for face training. Even then, final results still show that these components deliver a sound outcome.

5.2.2 Cost Function for Appearance Matching

Matching the 3D face morphable model to the given face images is a process of model parameter estimation, in which a bunch of coefficients involved are required to be determined. For example, camera and illumination model is adopted in the projection of

the 3D face model into the image plane since 3D face model and 2D input facial images cannot be measured directly. Aiming at retrieving a 3D face the closest projective image to the input facial image, the error function between 3D model projective image I_{mod} and input image I_{input} is described as:

$$E_I = \sum_{x,y} \left\| I_{input}(x,y) - I_{mod}(x,y) \right\|^2$$

(5.1)
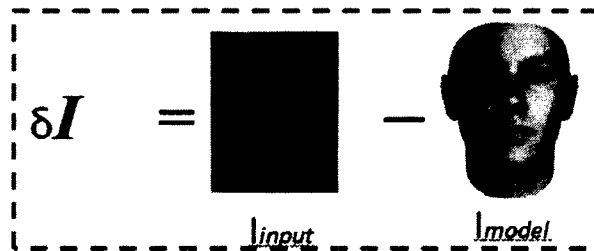
Fig. 5.8 Illustration to the difference between the initial image and the target

5.2.3 Fitting Process

To achieve a sound result from 3DMM estimation of human faces, we divide our algorithm into following steps.

1) Annotating landmarks on the target image.

2) Denoising the target image using image diffusion filtering.

3) Estimating the head orientation by fitting the feature points of the 3D model to those on the target image.

4) Facial appearance fitting estimation.

a) At the beginning phase, estimating the set of parameters for color matching dominates the optimization while estimating the feature vectors with respect to the principal components α and β has not yet begun.

b) Once the color of the face model runs close to the target image, α and β start to evolve into the optimization. DE populations are initialized by Gaussian distribution of $(0 \sim 1)$ instead of uniform distribution.

c) The optimization process runs iteratively until certain predefined criterion is completed. Figure below shows the learning curve of one of the learning cases.

5.2.4 Learning Face Model

In this section, we show our experimental results obtained over lighting and pose variations. The test images applied in our experiments are literally divided into two separate parts, frontal images and non-frontal images.

A. Frontal Image Test

We first use the frontal image from the ODU-VL face database as the training image. Figure 5.9 shows the original image and synthetic image of a test individual.

(a)   (b)

Fig. 5.9 Original image and synthesized image,
(a) original image; (b) synthesized image

(1) Lighting Variations

Once the 3D model is reconstructed, it is ready to re-render the 3D scene by using different lighting variations. Figure 5.10 shows composited scenarios dominated by light sources in different positions. Given constant distance from the light to the object, we change the light source positions by varying azimuth and elevation angles of the light source.
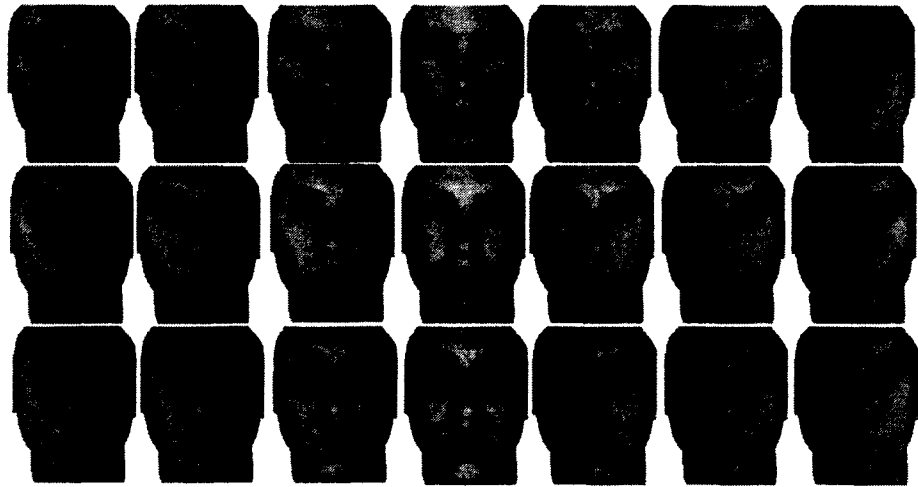
Fig. 5.10 Faces rendered under different illumination conditions;
azimuth and elevation angles ranging from -60 to 60 and -30 to 30 respectively.

(2) Pose Variations

Figure 5.11 shows composited scenarios rendered by various pose conditions. We diversify the face orientation by assigning different azimuth and elevation angles to the transform matrix of rotation.
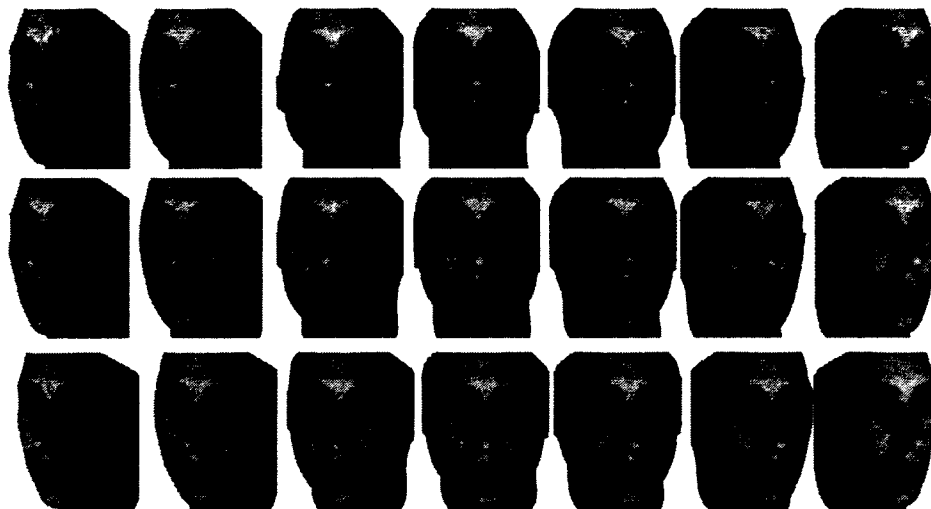
Fig. 5.11 Faces rendered under different pose conditions;
azimuth and elevation angles ranging from -30 to 30 and -10 to 10 respectively.
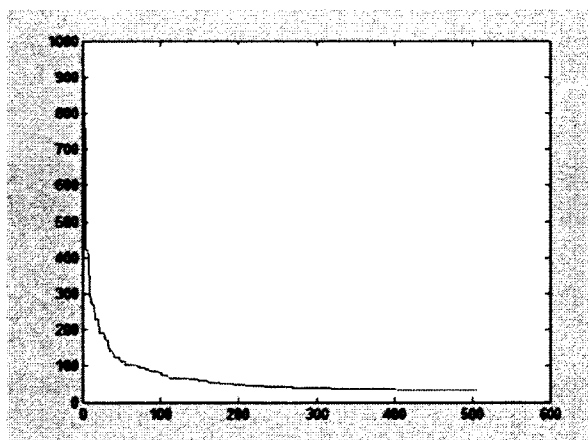
Fig. 5.12 Learning curve of facial estimation using 3DMM and DE

B. Non-Frontal Image Test

Most scenarios in real world applications contain non-frontal images. To demonstrate the feasibility of our framework on non-frontal images, we execute our experiments on the CMU-PIE database, from which we select the image with the out-of-plane rotation and controlled lighting source. Figure 5.13 depicts the original 2D image (a) used to generate

the 3D model and the associated synthesized image (c) as long as they are overlaid

composition (c).

(a) (b) (c)
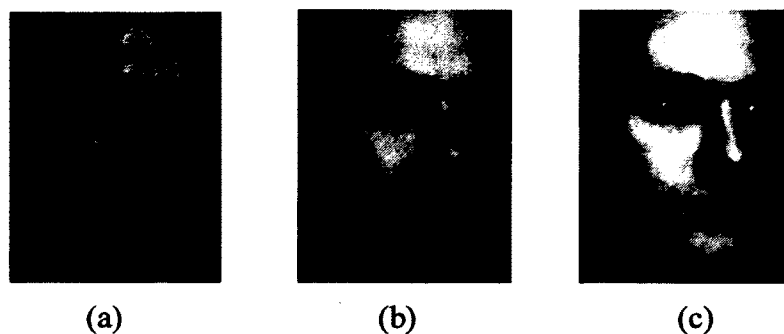
Fig. 5.13 Experimental results from non-frontal face training, (a) the original image; (b) the original image overlaid by the synthesized; (c) the synthesized image
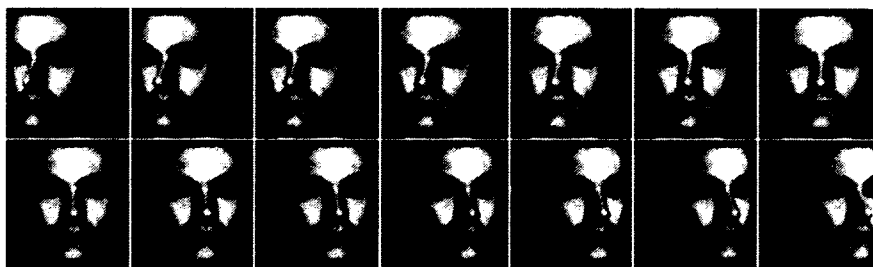
Fig. 5.14 Pose variations w.r.t. the 3D model generated in Figure 5.13

Figure 5.14 shows the pose variations based of the 3D face model generated above under the same illumination conditions. In order to make comparison of the experiments above, the new 3D face model is imposed by varied lighting directions. Figure 5.15 depicts the output of the proposed framework.

Fig. 5.15 Lighting variations w.r.t the 3D model generated in Figure 5.13

Please note that the facial appearance demonstrated on the generated 3D model is different from the original image. That reason could be explained from two aspects; the original image is dark and stained with noise which makes us preprocess the original image to improve its "*readability*"; Secondly, we use the first 70 features of the PCs to reconstruct the face model, which means the majority parts of the model has been reconstructed while partial information has been lost.

5.2.5 Validation for Synthesized Faces

To validate our algorithm, we apply our model on CMU-PIE database whose image gallery is collected from 68 subjects across 13 poses and under 43 illumination conditions [8]. We select images of 6 subjects, which are taken under 3 different lighting conditions and with 5 different pose orientations. We use the frontal image as training data for the 3D model generation, while the rest are included in the test dataset for validation. Figure 5.11 shows images used in our preliminary experiment.
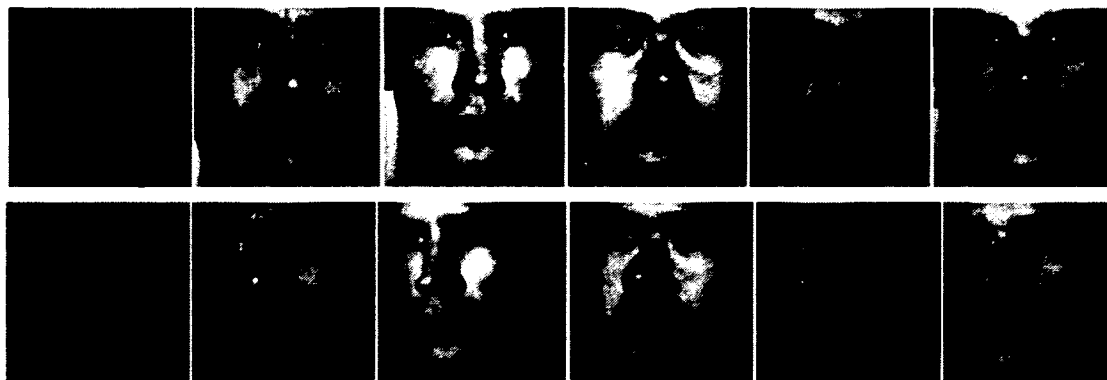
Fig. 5.16 Training and test images with different subjects

We utilize Principal Component Analysis (PCA) [55] to evaluate the synthesized images rendered per individual. Each test image is associated with a cluster of 9 synthesized 2D images that are included in the PCA training dataset for recognition purposes. An example of the test image and its corresponding training images are showed in Figure 5.17.
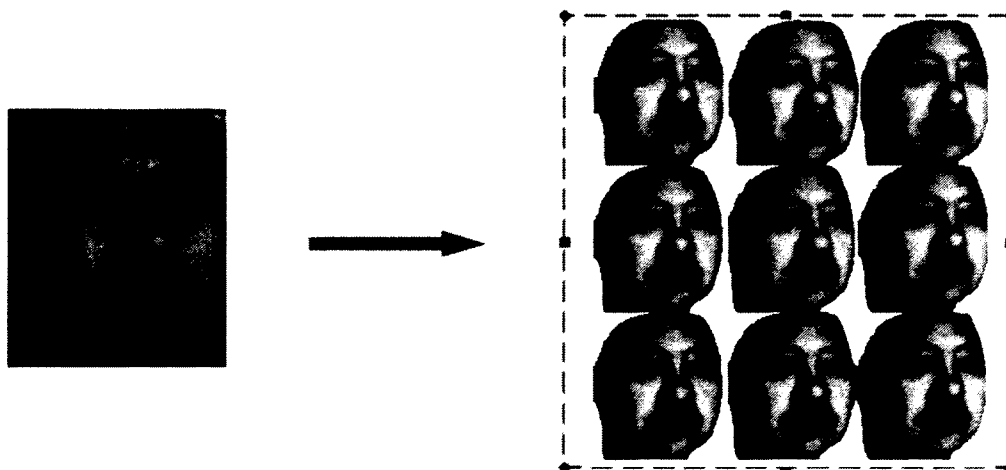
Fig. 5.17 Example test image and corresponding training images. The image in the center of clustered images (right) is imposed with same orientation as the real image (left). The surrounding images (right) are created by minor orientation offsets to the corresponding pose view, which are varied by azimuth and elevation angles ranging from -3 to 3 degrees.

Figure 5.18 shows the evaluation results from PCA with respect to different training datasets. The top curve (star) is obtained by using test images as training images. The bottom curve (diamond) informs the FR accuracy merely from frontal images training. The curve in the middle (circle) indicates the performance of training images projected by the synthesized 3D model. The comparison between these curves shows that the generated images, to some extent, achieve similarities to the real images. An illustration of the camera, flash lighting positions as well as head positions are plotted in Figure 5.19.
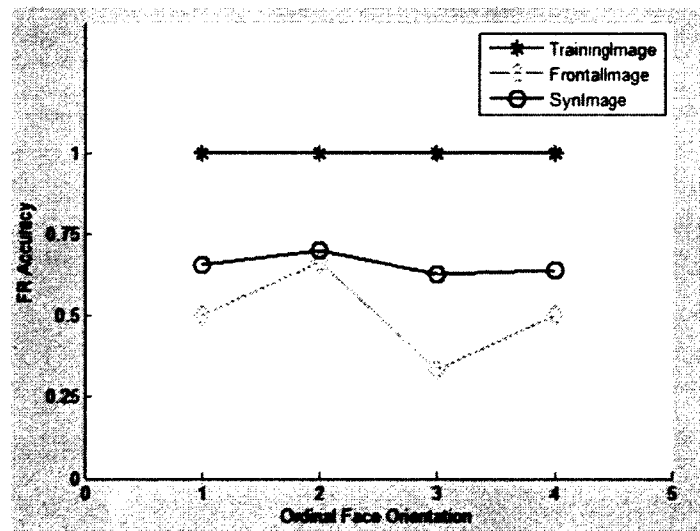
Fig. 5.18 Face orientations vs. face recognition accuracy w.r.t different sets of training images.
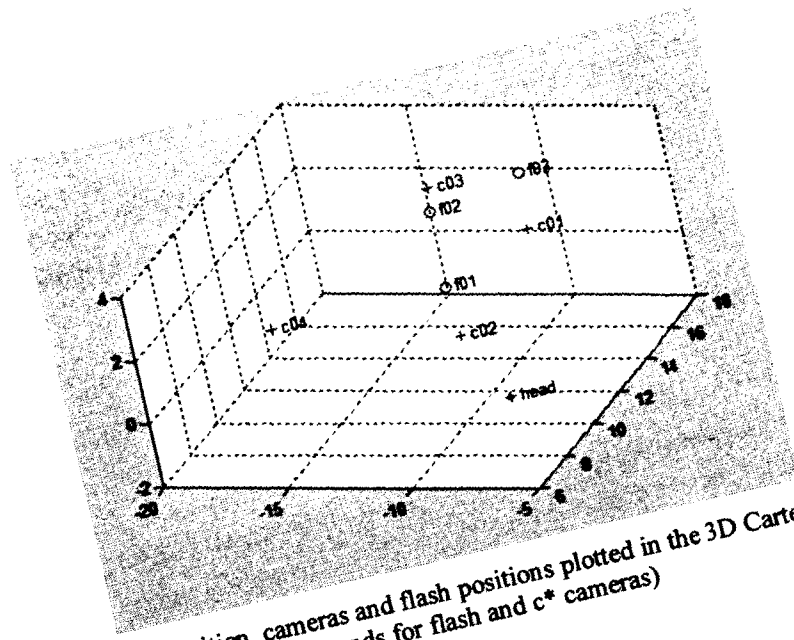
Fig. 5.19 The head position, cameras and flash positions plotted in the 3D Cartesian coordinate. (f* stands for flash and c* cameras)

CHAPTER 6

BACKWARD TEXTURE MAPPING

6.1 Introduction

In Chapter 4, we proved that it is feasible to reconstruct 3D face models from single 2D images. However, in some cases, the synthesized texture is not comparable to the real image. Consequently, the new database created cannot provide reliable training datasets to the face recognition system. Since the synthetic texture is incompatible to the database construction, we attempt to find our solution from the target image, which leads us to texture mapping and image warping techniques.

6.2 Texture Mapping

Texture mapping [56, 57] can be reached by two ways which include forward mapping and backward mapping. Forward mapping refers to projecting the texture pattern of the source image from the pixel coordinate onto the target image. Backward mapping indicates the inverse operation: mapping from the target image to the source image. As we collect the texture data from image pixels, we are supposed to choose the backward mapping. To better clarify the facial texture space, we denote it as $<U, V>$, which is illustrated in Figure 6.1. The discrepancy in data representation between texture space and image space is obvious. First, pixel value in image is regularized in grid data whereas vertex value in texture is scattered. Second, the facial appearance in image space is quite different from that in texture space. However, image warping makes the issue resolvable.
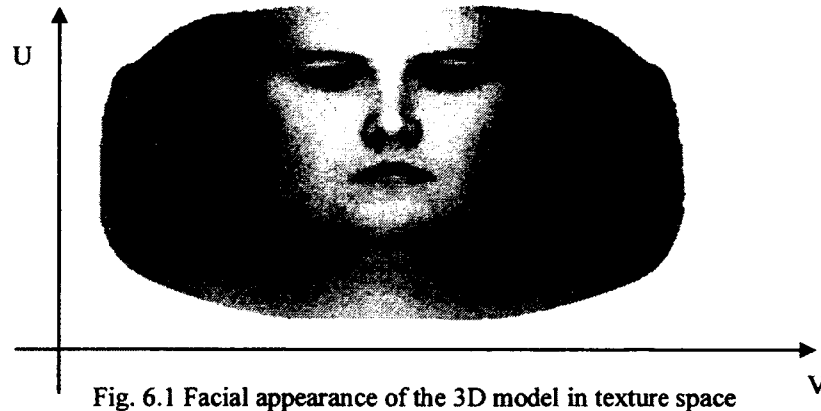
Fig. 6.1 Facial appearance of the 3D model in texture space

6.3 Image Warping

6.3.1 Background

Image warping [25, 58, 59] is regarded as a smooth transformation between the source image and the target image. In contrast to statistical-based model, it is an appearance-based image morphing technique that is more convenient and straightforward for implementation. Since it only concerns the pixel-level transformation in the image space, there is no necessary seeking prior knowledge from the training data. Given the correspondence geometric feature points marked on both images, a mapping function is established [60, 61]. From the viewpoint of *Shape* and *Texture*, as we discussed in the previous chapter, geometric feature points along with their topological connections could be regarded as *Shape* information; whilst the color blending or interpolation as the computation based on *Texture* information.

6.3.2 Previous works

It is naturally concluded from aforesaid that the mapping function plays an important role in image warping. Brookstein [62, 63] attempts a way of minimizing the amount of bending using thin-plate splines. Alexa et al. [64] introduced the concept of as-rigid-as-possible transformation to image morphing. At first, a mesh structure is associated to the image. Then they solve a linear system of equations whose size is equal to the number of vertices in the mesh to produce it as-rigid-as-possible deformation. They dissect source and target objects into isomorphic complexes. Igarashi et al. [63, 65] proposes a point-based image deformation technique for cartoon-like images in which the resulting deformations are as "rigid-as-possible." Such deformation has the property that the amount of local scaling and shearing is minimized. Schaefer et al. [66] addresses a 2D image deformation technique by seeking for the optimal rigid transformation that maps a set of weighted points which handles to their deformed positions, at each image domain point. Because the optimal transformation is calculated and assigned for every point on the image, this method is called image deformation based on Moving Least Square (MLS) [67, 68].

6.3.3 Image Deformation based on Moving Least Square (MLS)

Three properties make it more feasible to apply texture mapping.

1. Interpolation: the handles p on the source image should map directly to q under deformation.

2. Smoothness: deformation function is supposed to generate smooth deformation.

3. Identity: mapping function should be keeping identity when both handlers on the source image and target image are the same.

Schaefer et al. [66] construct a deformation function f based on the idea of Moving Least Square. Let us define p_i a handler in the image and q_i the corresponding deformed positions to p_i. Given a point v in the image, we try to find the best affine transformation $l_v(x)$ that minimizes the cost equation E_c,

$$E_c = \sum_i w_i \left| l_v(p_i) - q_i \right|^2 \tag{6.1}$$

p_i and q_i are both row vectors and the weights w_i are denoted as:

$$w_i = \frac{1}{\left| p_i - v \right|^{2\alpha}} \tag{6.2}$$

Since $l_v(x)$ is the affine transformation, it can be written in the form as following,

$$l_v(x) = xM + T \tag{6.3}$$

We define p* and q* be the weighted centroids of p_i and q_i respectively,

$$p^* = \frac{\sum_i w_i p_i}{\sum_i w_i} \quad , \quad q^* = \frac{\sum_i w_i q_i}{\sum_i w_i} \tag{6.4}$$

then a translation T yields,

$$T = q^* - p^* M \tag{6.5}$$

So $l_v(x)$ yields

$$l_v(x) = (x - p^*)M + q^* \tag{6.6}$$

which means equation (1) can be represented in the form as

$$E_c = \sum_i w_i \left| \bar{p}_i M - \bar{q}_i \right|^2 \tag{6.7}$$

where $\bar{p} = p_i - p*$ and $\bar{q} = q_i - q*$. Considering M is an affine transformation, we can conclude the deformation function $f_a(v)$ as:

$$f(v) = \sum_j A_j \bar{q}_j + q* \tag{6.8}$$

in which A_j is given by

$$A_j = (v - p*) \left(\sum_i \bar{p}_i^T w_i \bar{p}_i \right)^{-1} w_j p_j^T \tag{6.9}$$

Additionally, if M is a rigid transformation,

$$f(v) = VU^T (x - p*) + q* \tag{6.10}$$

where U and V can be obtained through Singular Value Decomposition, which is

$$U\Lambda V^T = \sum_i w_i(x)(p_i - p*)(q_i - q*)^T \tag{6.11}$$

6.3.4 Application of MLS Deformation

The MLS deformation is found to have a wide range of applications in image analysis, medical image processing, computer vision etc. Yuanchen Zhu and Steven Gortler [69] even extend the idea of MLS to the 3D settings to control 3D object deformation by manipulating a set of handlers. Figure 6.2 illustrates the comparison of the facial expressions before and after MLS deforming the input image (a). Note that there is a minor offset in (c) of the control point close to mouth right corner comparing to the same point in (b).
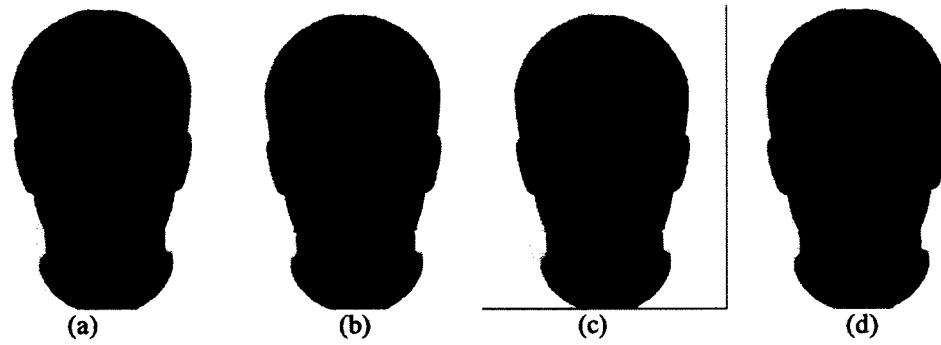
Fig. 6.2 Illustration of rigid transform of MLS deformation on faces
(a) the original neutral face, (b) control points,
(c) deformed position of control points, (d) smiling faces by deformation

We applied the MLS deformation to the texture mapping from the image space to the texture space. The control points p are manually selected from the image. The deformed points q are manually selected from the vertices in the texture space. As shown in Figure 6.3, once the deformation transformation f is established between the pixel on frontal face image and the texture vertices, the backward mapping is operated. The vertices index in the texture space is fully corresponding to those in shape space. This means the newly deformed texture is ready to impose on the 3D face surface.
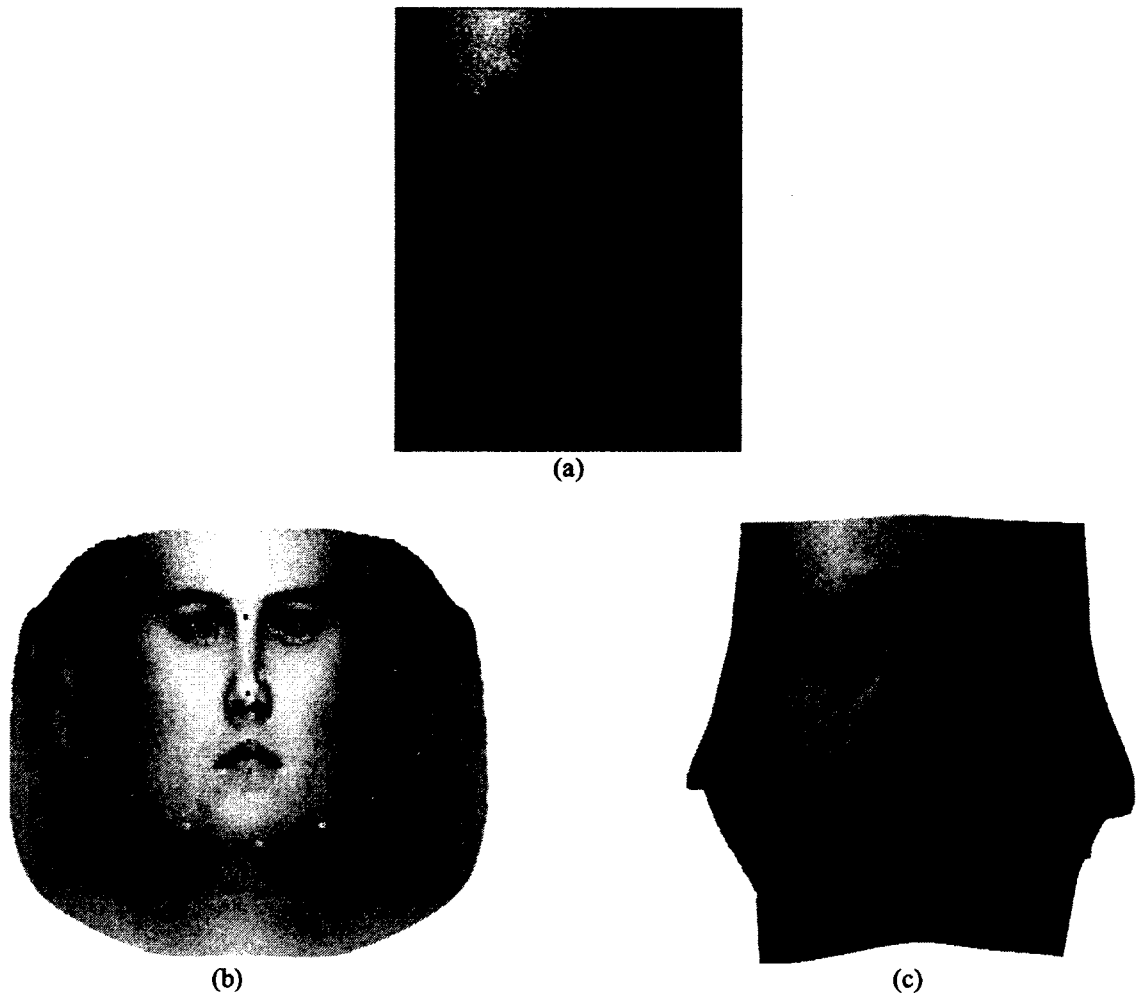
Fig. 6.3 MLS transformation from image space to texture space,
(a) frontal face image with control points p,
(b) texture space with deformed positions marked,
(c)newly generated image in texture space

Figure 6.4 shows the comparison between the synthesized results before (a) and after (b)

the 3D shape is imposed by the real image after the backward texture mapping.

(a)

(b)
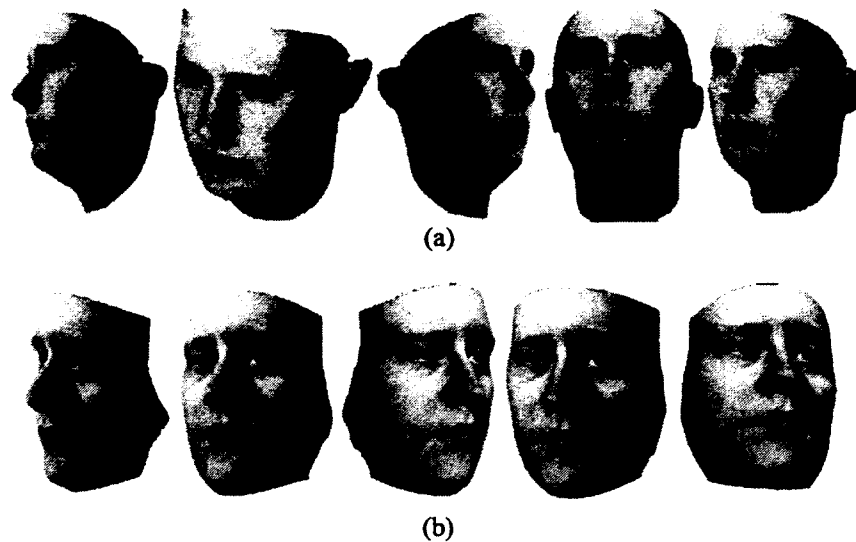
Fig. 6.4 Synthesized images before (a) and after (b) texture mapping

6.4 Image Warping Applied to 3D Face Synthesis

Since the aforementioned approaches are available for the synthesis of 3D faces, we thus attempt to implement them to the face dataset for more quantitative valuations. We apply our technique to the candidate images from CMU-PIE database, which aims at providing comparable results to those obtained in Chapter 5.

6.4.13D Faces Synthesis

Figure 6.5(b) shows the initial state of the 3D face that is overlaid on the original image. The training process consists of several stages whose major functionality is different from the others. The first stage, denoted as Stage I, is the process of 3D-2D rigid registration, in which the 3D transform functions the 3D face iteratively to fit to the pose of the target image. Besides the rigid registration, Stage I performs the color registration between the 3D face model and the target image. Figure 6.5(c) demonstrates the process of rigid shape registration and color matching.
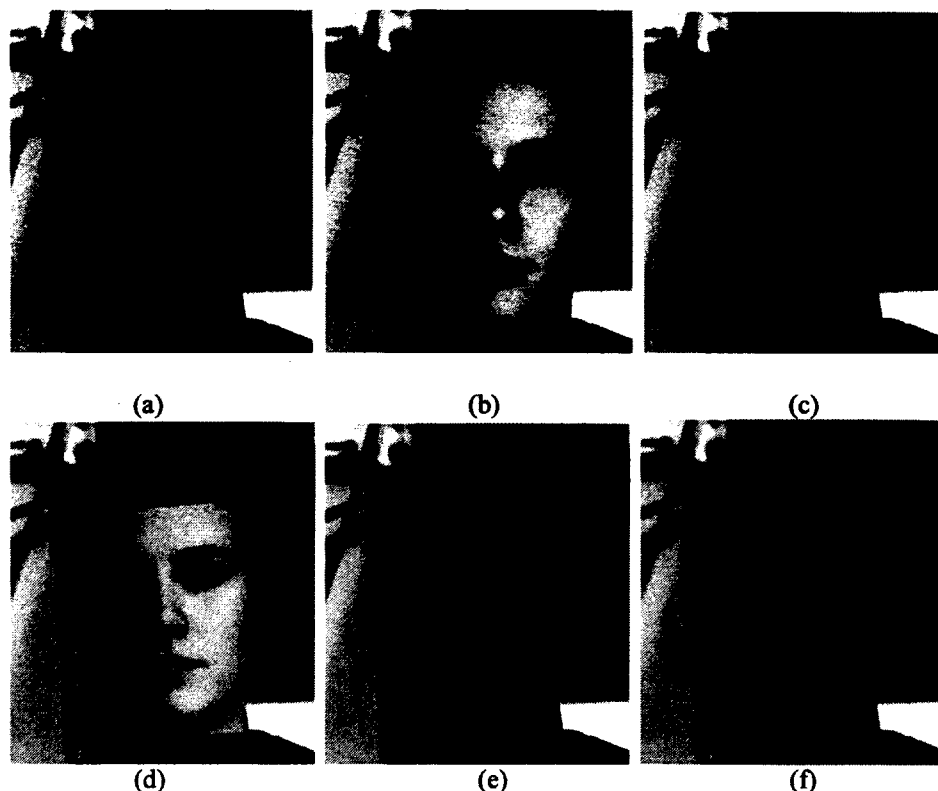
Fig. 6.5 Illustrations to different training stages; (a) original image, (b) initial state, (c) Stage I, (d) Stage II, (e) Stage III, (f) Stage IV

The second stage, denoted as Stage II, is the process of illumination matching by estimating the position of the light source in 3D spaces. In this stage, the majority contribution of the target image to the training is the specular highlights and the cast shadows. In our experiments, the Phong illumination model is employed to simulate the reflectance of the object surface. We present the illumination context with different lighting positions in Figure 6.5(d).

In Stage III, we start manipulating 3DMM to generate the morphed faces whose projections on 2D planes are used to measure the distance of the 3D face to the target. Figure 6.5(e) shows the result of the synthesized face overlaid on the target face.

In Stage IV, the image warping technique, MLS, is used to enhance the facial appearance of the 3D faces. Figure 6.5(f) shows the final result of the system. Please note that, due to the original image pre-processed prior to the morphing procedure starts, the associated color space presented on the synthesis faces is biased from the original image.

6.4.2 Validation of Synthesized Images

Following the same procedure in Chapter 5, we generate a training dataset that consists of images synthesized from 3D faces under different pose and illumination conditions.
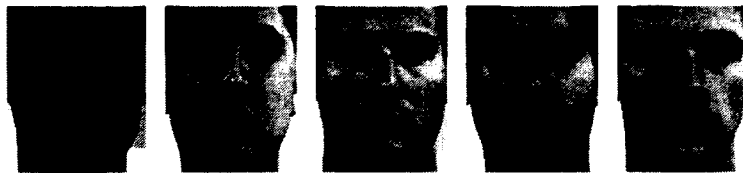
Fig. 6.6 Frontal images from five different subjects which are used for the training of 3D models

We select five candidates to create the corresponding 3D models based on their frontal images, denoted as $I_{tr,i}$ in which ($i=1,2,...5$). All five subjects are listed on the gallery in Figure 6.6. In addition, twelve images with different poses or lighting conditions from each individual are picked up as the test dataset, $I_{tst,i}$.

The projections of synthesized 3D faces on 2D planes are denoted as $M_{tr,i}$, which are used as the 2D face recognition training dataset. We tentatively create two sets of M_{tr} , M^C_{tr} and M^u_{tr} , which are produced from the controlled and uncontrolled parameters of pose or lighting imposers. The "controlled" parameters indicate that the light or pose coordinates has been provided by the database, while the "uncontrolled" parameters the random coordinates within certain ranges. In M^C_{tr} , we create three snapshots with minor

pose offsets for each image from $I_{tst,i}$. There are 36 M^u_{tr} snapshots and equal number of M^C_{tr} snapshots for each subject to ensure M_{tr} has the best coverage of the possible scenarios.

Figure 6.7 gives the comparison between the images in $I_{tst,l}$ and their associated synthetic images, in M^C_{tr}, projected under the same pose and lighting scenario. In Figure 6.8, we demonstrate some of the images in M^u_{tr}.
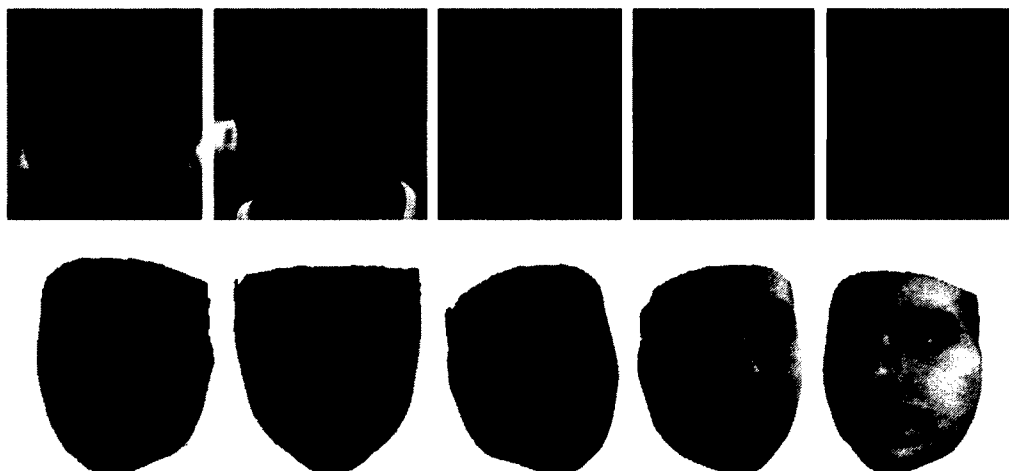
Fig. 6.7 Comparison between test faces and the virtual faces,
the images on the top are real images from the PIE database; the bottom row contains face images
generated under the controlled pose directions and light source positions

Fig. 6.8 Samples of images generated without the control on pose directions and light source positions

Figure 6.9 shows the learning curves of the 3D faces generation. Each individual is represented with different colors. It could be learned from the diagram that the error curve starts from significant differences between the mean face and the target face. Then the curve drops rapidly until they reach their 60~70 iterations. As the iterations increase, the optimization process marches closer to its optimum, which means the 3D face model finds its optimal linear combination to present the target face.
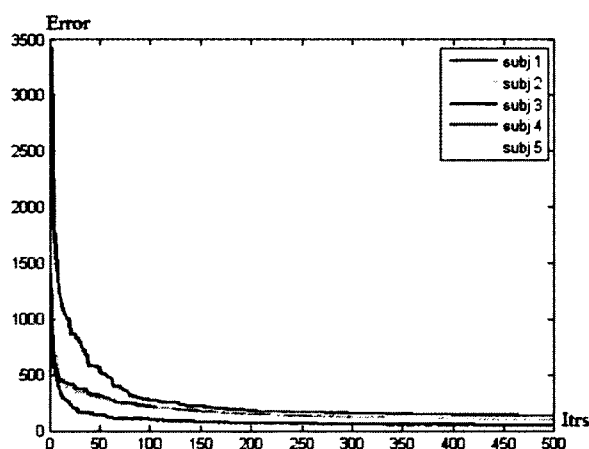
Fig. 6.9 Learning curve from five subjects with errors v.s. iterations

The bar graph in Figure 6.10 indicates the FR rate obtained under three different training sets. We still use PCA to execute the validation process. The blue bar indicates the FR rates without training but with frontal faces. The green bar indicates FR rates by using uncontrolled database in the training phase. The red bar depicts FR rates by using controlled database in the training phase. The plot preliminary proves the synthesized faces, to some extent, enhance the performance of the FR system. At the same time, those faces generated under controlled parameters bring more accuracy to the system than uncontrolled parameters.
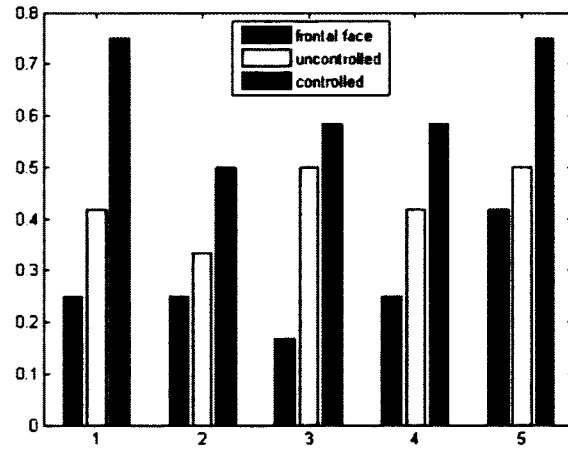
Fig. 6.10 Bar-plot of FR rate v.s. subjects

CHAPTER 7

CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

Several factors can effectively decide whether the performance of the face recognition system is compatible with practical applications, such as Homeland Security, real-time monitoring, access control, etc. On the one hand, various face recognition algorithms were invented in the last decades. As shown in [2, 55, 58], researchers continuously create innovative ideas to push face recognition advances forward to the next level; on the other hand, it is eagerly desired for as many training face images as possible to expand system-owned knowledge about potentially unknown human faces. A database that is manually created, to some extent, could be one of the solutions to this problem. However, in most cases, gathering sufficient data from candidates could be either impossible or intended for extra resource consumption.

Due to the aforementioned facts, the research topic of this paper is introduced into discussion. In this dissertation, we propose a novel framework for creation of large scale of 2D database from single image. In Chapter 2, we demonstrate an overview diagram of the whole framework.

In order to achieve that goal, we first attempt to reconstruct the 3D model from the input image and then project the 3D faces model onto 2D image space. In our work of 3D model reconstruction, the 3D Morphable Model is employed to estimate the 3D shape and texture of the incoming face. Theoretically speaking, 3DMM is able to synthesize face

images under pose and illumination variations. This technique can be used in either face analysis of model generation from images or to generate training or test images at any imaging condition [31]. Hence, it can be regarded as the model for a meta-database that allows the creation of augmented images for test or training purposes.

Considering the complicated context of the incoming images could be, we include into the estimation process several parameters with respect to shapes, textures, shadows and light directions. The estimation process of fitting shape and texture information into 2D images is considered as the problem of searching for the global minimum in a high dimensional feature space, in which optimization is apt to have local convergence.

In Chapter 4, we introduce the concept of global optimization and how it is related to our work. One of the most important members of the global optimization algorithms is Evolutionary Algorithms whose basic idea derives from the inspiration of biological mechanisms. The Differential Evolution algorithm is regarded as a powerful optimizer in the EAs family. Hence we integrate DE into the framework to tackle the local optimization. The experimental results, in Chapter 5, demonstrate that we are able to automatically create a virtual 3D face from a single 2D image with high performance.

3DMM is a statistical-based model in which the synthesized pixels from the model are estimated according to the likelihood of their existence. Additionally, as the MSE is used in our global cost function, the synthesized face textures are shown to be characterized as focusing more on the global appearance instead of local facial features. In order to generate realistic texture information, in Chapter 6, we integrate the backward texture mapping to the 3D face reconstruction that includes the image warping technique,

Moving Least Square Image Deformation. In our work, MLS deformation is used to construct the mapping function from the image space to the texture space. Our results have shown the beneficial effect brought by the texture mapping to the synthesized 3D model as well as the generated 2D face database.

7.2 Future Works

Future research will focus on the following sections:

1. Reduce Time Consumption

Aforementioned in Chapter 3, Differential Evolution is employed to keep the global optimization from the local minimum. However, the computation efficiency of DE optimization would be a considerable issue. The future work should be more emphasized on the decrease of computing time.

2. Pre-processing on input images

The facial appearance in the input image is undetermined. One of the disadvantages brought by this fact is that in some cases, it is difficult for the 3D reconstruction scheme to estimate satisfactory results. This is because only the pixel-level comparisons of intensity image are used in the cost function. In the future work, more constraints are suggested to be included like illumination and reflectance to retain the result from divergent away from the optimal point.

3. Input Image Annotations

Either in the process of facial estimation or texture mapping, landmark annotation plays an important role in our implementation. Nevertheless manual annotations are both time consuming and involve human interactions, which is an unstable factor. Therefore, future work is recommended in the direction toward an automatic feature points annotation.

4. Database Simplification

BFM database is featured as a high resolution 3D faces. In other words, it provides high definition of the faces but, on the other hand, uses over 60k vertices to represent human faces. Practically speaking, it is not necessary to apply 3D faces about this level to lower resolution images. If we regard the computational complexity as one of the factors that affect the face reconstruction, it is highly recommended to downsample face models to a low resolution with few number of vertices like 5,000 ~6,000.

5. 3D Representation of Face Expressions and Movements

The 3D reconstruction is able to be working on single 2D image or multiple 2D images with neutral facial expressions. The future works with applications on multiple expressions is supposed to be one of the research directions, which means the facial movements is able to be represented and reconstructed on basis of certain sets of features. One possible solution is using the synthesis of 3D face database and Structure from Motion (SfM) [60] techniques to mimic the movement of facial key features.

REFERENCES

[1]   P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, "Face recognition vendor test 2002," in *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, 2003, p. 44.

[2]   W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, pp. 399-458, 2003.

[3]   Y. Adini, Y. Moses, and S. Ullman, "Face Recognition: The Problem of Compensating for Changes in Illumination Direction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 721-732, 1997.

[4]   A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 643-660, 2001.

[5]   S. U. C. S. Dept, T. G. Dietterich, and B. G. Buchanan, *The role of the critic in learning systems*: Computer Science Dept., Stanford University, 1981.

[6]   J. R. Quinlan, "Induction of Decision Trees," *Mach. Learn.*, vol. 1, pp. 81-106, 1986.

[7]   S. Z. Li and A. K. Jain, *Handbook of face recognition*: Springer, 2005.

[8]   T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) database," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, 2002, pp. 46-51.

[9]   A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters,* vol. 28, pp. 1885-1906, 2007.

[10]   M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision,* vol. 1, pp. 321-331, 1988.

[11]   M. Stegmann and D. Gomez, *A brief introduction to statistical shape analysis,* 2002.

[12]   S. Baker, I. Matthews, and J. Schneider, "Automatic Construction of Active Appearance Models as an Image Coding Problem," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 26, pp. 1380-1384, 2004.

[13]   G. Xinbo, S. Ya, L. Xuelong, and T. Dacheng, "A Review of Active Appearance Models," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on,* vol. 40, pp. 145-158, 2010.

[14]   I. Matthews, J. Xiao, and S. Baker, "2D vs. 3D Deformable Face Models: Representational Power, Construction, and Real-Time Fitting," *Int. J. Comput. Vision,* vol. 75, pp. 93-113, 2007.

[15]   R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof, "Fast Active Appearance Model Search Using Canonical Correlation Analysis," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 28, pp. 1690-1694, October 2006.

[16]   R. Beichel, G. Gotschuli, E. Sorantin, F. W. Leberl, and M. Sonka, "Diaphragm dome surface segmentation in CT data sets: a 3D active appearance model approach," presented at the Medical Imaging 2002: Image Processing, 2002.

[17] R. Beichel, S. Mitchell, E. Sorantin, F. Leberl, A. Goshtasby, and M. Sonka, "Shape- and Appearance-Based Segmentation of Volumetric Medical Images," presented at the Proc Int Conf on Image Processing (ICIP), 2001.

[18] J. J. Atick, P. A. Griffin, and A. N. Redlich, "Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images," *Neural Computation,* vol. 8, pp. 1321-1340, 1996.

[19] J. Lee, B. Moghaddam, H. Pfister, and R. Machiraju, "Silhouette-based 3D face shape recovery," *Graphics Interface,* 2003.

[20] S. Romdhani and T. Vetter, "Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior," 2005, pp. 986-993 vol. 2.

[21] A. Patel and W. A. P. Smith, "3D morphable face models revisited," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on,* 2009, pp. 1327-1334.

[22] I. Kemelmacher-Shlizerman and R. Basri, "3D Face Reconstruction from a Single Image Using a Single Reference Face Shape," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 33, pp. 394-405, 2011.

[23] D. Beymer and T. Poggio, "Face recognition from one example view," Cambridge, MA, USA, 1995, pp. 500-507.

[24] S. Ullman and R. Basri, "Recognition by Linear Combination of Models," Massachusetts Institute of Technology1989.

[25] D. Ruprecht and H. Muller, "Image warping with scattered data interpolation," *Computer Graphics and Applications, IEEE,* vol. 15, pp. 37-43, 1995.

[26]   T. Vetter and T. Poggio, "Linear object classes and image synthesis from a single example image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 733-742, 1997.

[27]   T. F. Ezzat, "Example-Based Analysis and Synthesis for Images of Human Faces," *MIT EECS Masters Thesis,* 1996.

[28]   P. Breuer, K.-I. Kim, W. Kienzle, B. Scholkopf, and V. Blanz, "Automatic 3D face reconstruction from single images or video," in *Automatic Face & Gesture Recognition, 2008. FG '08. 8th IEEE International Conference on,* Amsterdam, Netherlands, 2008, pp. 1-8.

[29]   V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," in *Siggraph 1999, Computer Graphics Proceedings,* 1999, pp. 187-194.

[30]   V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 25, pp. 1063-1074, 2003.

[31]   P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," in *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on,* 2009, pp. 296-301.

[32]   C. A. Floudas and C. E. Gounaris, "A review of recent advances in global optimization," *Journal of Global Optimization,* vol. 45, pp. 3-38, 2009.

[33]   R. Storn and K. Price, "Differential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces," *INTERNATIONAL COMPUTER SCIENCE INSTITUTE-PUBLICATIONS-TR,* 1995.

[34] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308-313, 1965.

[35] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, pp. 4-31, 2011.

[36] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *Evolutionary Computation, IEEE Transactions on*, vol. 9, pp. 303-317, 2005.

[37] T. Back, U. Hammel, and H. P. Schwefel, "Evolutionary computation: comments on the history and current state," *Evolutionary Computation, IEEE Transactions on*, vol. 1, pp. 3-17, 1997.

[38] D. Zaharie, "Critical values for the control parameters of differential evolution algorithm," in *Proceedings of MENDEL 2002, 8th International Mendel Conference on Soft Computing, June 2002, Brno, Czech Republic*, 2002.

[39] V. Feoktistov, *Differential Evolution: In Search of Solutions*: Springer US, 2006.

[40] R. Joshi and A. C. Sanderson, "Minimal representation multisensor fusion using differential evolution," presented at the Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997.

[41] J. Chiou, C. Chang, and C. Su, "Ant Direction Hybrid Differential Evolution for Solving Large Capacitor Placement Problems," *Power Systems, IEEE Transactions on* vol. 19, pp. 1794-1800, 2004.

[42]   J. Sander, M. Ester, H. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The Algorithm {GDBSCAN} and Its Applications," *Data Mining and Knowledge Discovery,* vol. 2, pp. 169-194, 1998.

[43]   D. K. Tasoulis, V. P. Plagianakos, and M. N. Vrahatis, "Differential Evolution Algorithms for Finding Predictive Gene Subsets in Microarray Data," presented at the AIAI, 2006.

[44]   N. Karaboga and B. Cetinkaya, "Performance comparison of genetic and differential evolution algorithms for digital FIR filter design," presented at the Proceedings of the Third international conference on Advances in Information Systems, Izmir, Turkey, 2004.

[45]   J. Brest, V. Zumer, and M. S. Maucec, "Self-Adaptive Differential Evolution Algorithm in Constrained Real-Parameter Optimization," in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on,* 2006, pp. 215-222.

[46]   V. Plagianakos, D. Tasoulis, and M. Vrahatis, "A Review of Major Application Areas of Differential Evolution Advances in Differential Evolution." vol. 143, U. Chakraborty, Ed., ed: Springer Berlin / Heidelberg, 2008, pp. 197-238.

[47]   R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization,* vol. 11, pp. 341-359, 1997.

[48]   C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," *Technical Report,* p. 17, 2010.

[49] D. Keren, M. Osadchy, and C. Gotsman, "Antifaces: A Novel, Fast Method for Image Detection," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 23, pp. 747-761, July 2001.

[50] R. Feraud, O. J. Bernier, J.-E. Viallet, and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 23, pp. 42-53, 2001.

[51] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 12, pp. 629-639, 1990.

[52] G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz, "Nonlinear anisotropic filtering of MRI data," *IEEE Transactions on Medical Imaging,* vol. 11, pp. 221-232, 1992.

[53] N. F. Troje and H. H. Bülthoff, "Face recognition under varying poses: The role of texture and shape," *Vision Research,* vol. 36, pp. 1761-1771, 1996.

[54] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," in *AVSS '09: Proceedings of the 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance,* 2009, pp. 296-301.

[55] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[56] F. M. Weinhaus and V. Devarajan, "Texture mapping 3D models of real-world scenes," *ACM Comput. Surv.,* vol. 29, pp. 325-365, 1997.

[57] P. Thevenaz, T. Blu, and M. Unser, "Interpolation revisited," *Medical Imaging, IEEE Transactions on,* vol. 19, pp. 739-758, 2000.

[58]   G. Wolberg, "Image morphing: a survey," *The Visual Computer,* vol. 14, pp. 360-372, 1998/12/01 1998.

[59]   M. Alexa, "Recent Advances in Mesh Morphing," *Computer Graphics Forum,* vol. 21, pp. 173-198, 2002.

[60]   F. L. Bookstein, "Size and shape spaces for landmark data in two dimensions (with discussion)," *Statistical Science,* vol. 1, pp. 181-242, 1986.

[61]   F. Bookstein, *Morphometric tools for landmark data : geometry and biology*: Cambridge University Press, 1997.

[62]   F. L. Bookstein, "Principal Warps: Thin-Plate Splines and the Decomposition of Deformations," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 11, pp. 567-585, 1989.

[63]   T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Trans. Graph.,* vol. 24, pp. 1134-1141, 2005.

[64]   M. Alexa, D. Cohen-Or, and D. Levin, "As-rigid-as-possible shape interpolation," presented at the Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000.

[65]   T. Igarashi and Y. Igarashi, "Implementing As-Rigid-As-Possible Shape Manipulation and Surface Flattening," *J. Graphics, GPU, & Game Tools,* vol. 14, pp. 17-30, 2009.

[66]   S. Schaefer, T. McPhail, and J. Warren, "Image deformation using moving least squares," *ACM Trans. Graph.,* vol. 25, pp. 533-540, 2006.

[67]   D. Levin, "The Approximation Power of Moving Least-Squares," *Mathematics of Computation,* vol. 67, pp. 1517-1531, 1998.

[68]   D. Levin, "Stable integration rules with scattered integration points," *Journal of Computational and Applied Mathematics,* vol. 112, pp. 181-187, 1999.

[69]   Y. Zhu and S. J. Gortler, "3D Deformation Using Moving Least Squares," Harvard Computer Science Technical Report: TR-10-072007.

APPENDIX A

Viola-Jones Face Detection

A.1. The Integral Image

For an incoming image, an integral image is represented as the sum of the pixels above and to the left of x, y, inclusive,

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{A.1}$$

in which $ii(x, y)$ is the integral image and $i(x, y)$ is the original image.

It is obvious to compute the sum of the pixels in the region of the rectangle (shadowed) which is shown in Figure A.1.

$$\sum_{(x,y) \in ABCD} i(x, y) = ii(D) + ii(A) - ii(B) - ii(C) \tag{A.2}$$

Figure A.2 shows four types of Haar-like features whose basic size if 24*24 pixels. Each of four features is shifted and scaled across all possible combinations, which means there are estimated 160,000 possible features to be calculated.
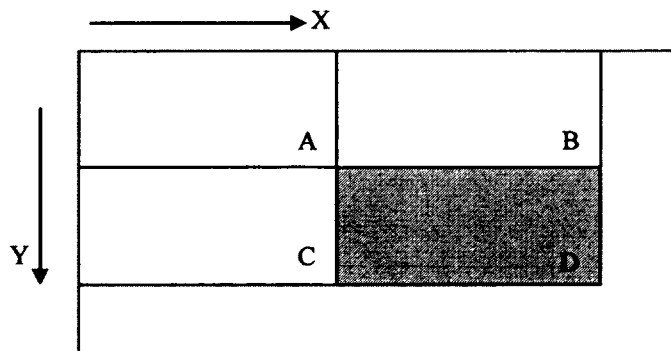

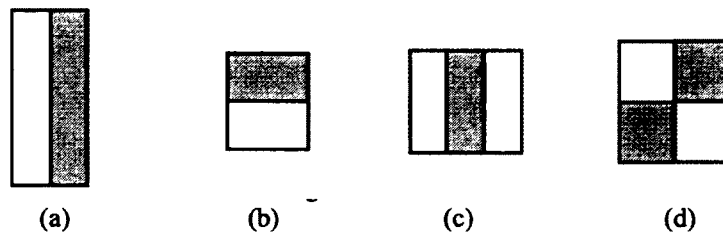Figure A.1 Illustration of calculation of the integral image

(a)   (b)   (c)   (d)

Figure A.2 Haar-like feature used in feature extraction

A.2 Adaboost Learning

Adaboost (Adaptive Boost) algorithm is regarded as a strong classifier, noted as $f(x)$, which is represented as the linear combination of *simple* and *weak* classifiers, noted as $h(x)$. Let's assume there are T weak classifiers involved in the classification. The final decision made by the algorithm is determined by the combination of the weighted weak classifiers.

$$f(x) = \sum_{t=1}^{T} a_t h_t(x)$$

(A.3)

Following is the pseudo code for the Adaboost algorithm:

I. Input

(1) Training examples $S = \{(x, z), i = 1, ..., N\}$;

(2) T is the number of weak classifiers;

II. Initialization:

(3) Initialize example score $F^0(x_i) = \ln(N_+/N_-)/2$, where N_+ and N_- are the number of positive and negative examples in the training data set.

III. Learning:

For $t = 1, \ldots, T$:

1). For each Haar-like feature $h(x)$ in the pool, find the optimal threshold H and confidence score c_1 and c_2 to minimize the Z score L^t;

2). Select the best feature with the minimum L^t;

3). Update $F_t(x_i) = F_{t-1}(x_i) + f_t(x_i)$, $i=1,\ldots, N$;

4). Update W_{+1j}, W_{-1j}, $j = 1, 2$.

IV. Output: $F^T(x)$.

A.3 Cascade of Boosted Classifiers

Cascade of boosted classifiers are regarded as the degenerate decision tree, which is featured as fast evaluation and quick rejection of sub-windows when testing. Each node in the structure shown in Figure A.3 is trained by using the false positives of the prior. In the method of Viola-Jones, they present a method of iteratively building boosted nodes to reach the desired false positive rate.
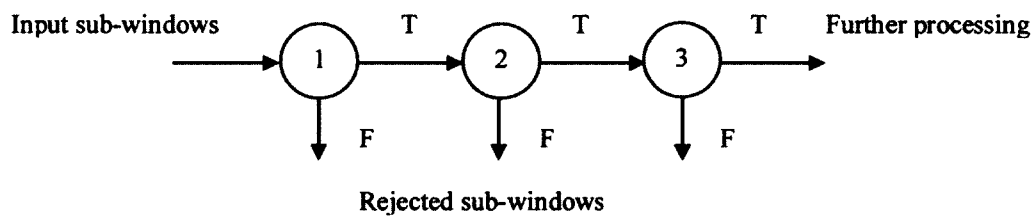

Figure A.3 Cascade of boost classifiers

VITA

Qun Wang

320 Suburban Pkwy.

Norfolk, VA 23505

Tele: (757)8397389

E-mail: qwang007@odu.edu

Education

Ph.D. in Electrical and Computer Engineering	**May 2012** (Expected)
Old Dominion University (U.S.)	09/2006 ~ Present
Master of Science in Electronic Engineering	**Mar. 2004**
Xidian University (P.R.China)	08/2001 ~ 03/2004
Bachelor of Science in Electronic Engineering	**Jun. 2000**
Shandong University (P.R.China)	09/1996 ~ 07/2000

Research Experiences

Research Assistant	**09/2006 ~ Present**
Vision Lab, Old Dominion University (U.S.)	
Computer Vision, Pattern Recognition	
Research Assistant	**08/2001 ~ 03/2004**
Xidian University (P.R.China)	
Digital Signal Processing, Image Processing	