Universität Karlsruhe
Fakultät für Informatik
Institut für Theoretische Informatik (ITI)
Prof. Dr. A. Waibel

WS 2006/07

Studienarbeit

Local Appearance-based 3D Face Recognition

Hua Gao

November 2006

Betreuer: M.Sc. H. K. Ekenel
Dr.-Ing. R. Stiefelhagen
Prof. Dr. A. Waibel

Table of Contents

Abstract

In this work,we present a local appearance-based approach for 3D face recognition. In the proposed algorithm, we first registered the 3D point clouds to provide dense correspondence between faces. Afterwards, we analyzed two mapping techniques, the closest-point mapping and the ray-casting mapping, to construct depth images from the corresponding well-registered point clouds. The depth images so obtained are then divided into local regions on which a discrete cosine transformation was performed to extract local information. The local features were combined at the feature level for classification. Experimental results on the FRGC version 2.0 face database showed that the proposed algorithm performs superior to the well-known face recognition algorithms.

Chapter 1

Introduction

Biometric identification is a challenging task that has received significant amount of interest for the last decades. Among the utilized biometric modalities, the human face is one of the most natural. Moreover, a subject's face images can be acquired easily and unobtrusively. Due to the low cost and wide availability of image acquisition systems, most of the face recognition algorithms are based on 2D intensity images [36]. However, the algorithms that process intensity images suffer from facial appearance variations that are caused by changes in head pose and illumination conditions. Much effort has been devoted to solving these problems in the 2D domain. Although significant enhancements have been achieved in the 2D domain against these variations under controlled conditions, the problem still remains unsolved under uncontrolled, real world conditions.

To handle the pose and illumination variations, utilizing 3D shape information has been shown to be a promising approach [6], since a point cloud or surface can represent the geometric structure of the face, and is not affected by pose variations and by extrinsic source of variations, such as illumination. A large number of approaches have been proposed for 3D face recognition. These approaches can be divided into three categories.

The first type of approaches consists of algorithms that only use the 3D shape information to extract features. In [18], Gordon used curvature information to describe face shape features. Principal curvatures and directions are used to extract the location of the nose and the eyes. The direction of the nose and faces are classified according to their volumetric difference resulting from a cylindrical ray tracing operation.

Another shape based face representation approach was applied by Tanaka et al. to represent facial surfaces [31]. In their work, the faces are represented with enhanced Gaussian images (EGI) based on maximum and minimum principal directions. Two EGIs representing rigid and valley lines, are extracted from facial surfaces and the identification is based on the matching of the extracted EGIs.

Chua et al. presented a powerful shape descriptor called point signatures in [9], which treat the face recognition problem as a non-rigid object recognition problem.

This algorithm, which was claimed to be invariant to facial expression changes, starts operating by first finding rigid facial points with the use of point signatures. Signature of all points of a test image are compared with those of training images and a coarse correspondence is established to feed the corresponding subset of points to a ranking and face registration step. In the ranking step, the models that have the most similar signatures of the points are selected for a fine registration with the iterative closest point (ICP) algorithm. This approach gets high accuracy in a small data set as they reported, but it requires high computational cost to compute point signature for each point.

In [25, 26], Lu et al integrated the ICP based rigid matching together with the non-linear thin plate spline (TPS) deformation. Feature vectors extracted from displacement vector field after deformation are classified with support vector machines (SVM) [34]. Decision is made by combining the rigid matching distance and the deformation classification result.

İrfanoğlu et al. also used ICP alignment to estimate landmarks automatically, and faces were recognized with their point set distance (PSD) technique [24]. Although experiments in [1] show that TPS-based registration may have side effects in terms of discrimination, TPS for intra-subject and inter-subject non-rigid deformations may increase performance in the case of expression variations [26].

The second type of 3D face recognition approaches represents 3D shape information in the 2D domain. In the depth-map based approaches [18, 32], 3D face is first translated and rotated in such a way that, it is directly frontal to the viewer and the z-axis is orthogonal to the view plane. The brightness of each pixel is directly proportional to the z-coordinate of the corresponding vertex. In this manner, vertices that are closer to the viewer are assigned a higher intensity value and vertices further away from the viewer are darker. By obtaining the so called depth-map image, one can apply the feature extraction technologies that are commonly used in the 2D domain, such as eigenfaces or fisherfaces [22, 30].

Three dimensional shape information can also be combined with 2D data as the third type of approaches to improve the classification performance. In [32, 8], shape information is represented with depth-map PCA coefficients and the texture information with PCA coefficients and the fusion is made at the decision level. It has been shown that the combination of these multi-modalities improves the recognition performance. In [35], Gabor wavelets are used to extract features from salient facial locations and point signatures are used to extract features from the corresponding points in the 3D shape. The fusion is made at the feature level and it is shown that this increases the recognition performance.

Bronstein et al. followed an approach based on geometric 3D invariants [7]. With the use of these geometric invariants, they mapped the 2D facial texture onto special 3D shape images, which are then fed into a PCA analysis. They stated in their work that this type of incorporation outperforms standard fusion of 2D images and depth-map images. For a detailed recent survey of 3D face recognition please see [6].

In this work, we present a novel 3D face recognition algorithm that is based on local

appearance face recognition. In the proposed algorithm, we first register the input point cloud in order to provide dense correspondence between the faces. Afterwards, we analyze two mapping techniques, the closest-point mapping and the ray-casting mapping to construct the depth images from the corresponding well-registered point clouds. Finally, we perform local appearance face recognition on these depth images.

Local appearance-based face recognition has been shown to be a promising approach for face recognition from intensity images. It is proposed as a fast and generic approach [15, 16] and does not require detection of any salient local regions, such as eyes, as in the modular or component based approaches [21, 29]. The approach has been tested extensively on the publicly available face databases and compared with the other well-known face recognition approaches. The experimental results showed that the proposed local appearance-based approach performs significantly better than the traditional face recognition approaches [15, 16]. Moreover, this approach is used for face verification on face recognition grand challenge (FRGC) version 1 data set [14], and it provided better and more stable results than the baseline face verification system. The approach is also tested under video-based face recognition evaluations and again provided better results [12, 13].

The remainder of the thesis is organized as follows. In Chapter 2, we describe the related algorithms we used in this study. Then we explain 3D face shape registration and depth image generation techniques as well as local appearance based 3D face recognition algorithm in Chapter 3. Experimental results are presented and discussed in Chapter 4. Finally, in Chapter 5, conclusions are given.

Chapter 2

Methodology

Automatic face recognition is the process of identification of an individual from his/her face by a computer. A typical face recognition system consists of three parts: face registration, face representation and classification. In this chapter, we will explain the algorithms that are used in this work.

2.1 Registration

Three dimensional data generally have different translation, rotation and scale parameters due to the position and orientation of the data acquisition device or to the pose variations of the subjects. One way to deal with these problems is to extract 3D features that are invariant to these transformations. Another approach is to first transform the shapes so that they have the same translation, rotation and scale parameters and then operate on the transformed shapes. Registration is a general term which means the process of finding a set of transformation matrices that will align several shapes into a common coordinate framework. The details and the mathematical formulations of ICP, one of the most widely used registration technique, will be given in the following section.

2.1.1 Iterative Closest Point (ICP) Algorithm

Iterative Closest Point (ICP) algorithm, proposed by Besl et al. [4], is a popular technique for both 2D and 3D shape registration problems. This algorithm can be used not only in biometric identification domain, but also in different areas such as surgery simulation and computer aided design (CAD) for objects modeling.

In order to match these point clouds of faces, we should transform faces to a common framework. The transformation we concerned in ICP is a rigid transformation, which includes a rotation and a translation. A point p in original shape is transformed

by multiplying a rotation R matrix and adding a translation vector t as described in equation (2.1.1).

$$p' = Rp + t. \tag{2.1.1}$$

So the main task of this algorithm is to compute the corresponding transform parameters iteratively. Each iteration contains three steps as summarized in Fig. 2.1.

In the first step, for each point p_i on a shape P, we will try to find the closest point p'_i on a common mesh X with Euclidean distance metric :

$$d(p'_i, p_i) = \|p'_i - p'_i\| = \sqrt{(x'_i - x_i)^2 + (y'_i - y_i)^2 + (z'_i - z_i)^2},$$

where $p_i = (x_i, y_i, z_i)$ and $p'_i = (x'_i, y'_i, z'_i)$ are two points in 3D space. If we have N_p point in point cloud P, the ICP algorithm will find the nearest neighbor to mesh X for all these points at the beginning of each iteration step. The corresponding closest points construct a new point cloud $P'_k = C(P_k, X) \subseteq X$, where the index k indicates the current iteration step of the registration.The original point cloud P is represented with P_0.

Then in the second step, the original shape P will be aligned to P' with transformation Q in a least-square sense. The ICP algorithm uses a quaternion based rotation representation, because quaternion is an efficient mathematical tool for formulating the composition of arbitrary spatial rotations. A quaternion $\dot{q}$ can be considered as a vector with four components or as complex number with three different imaginary parts:

$$\dot{q} = q_0 + iq_x + jq_y + kq_z,$$

A unit quaternion is therefore a quaternion of size one, $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$, where $q_0 \geq 0$. A rotation matrix R that uses a unit quaternion may be defined in such a way:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_0 q_z) & 2(q_x q_z + q_0 q_y) \\ 2(q_x q_y + q_0 q_z) & q_0^2 + q_y^2 - q_x^2 - q_z^2 & 2(q_y q_z - q_0 q_x) \\ 2(q_x q_z - q_0 q_y) & 2(q_y q_z + q_0 q_x) & q_0^2 + q_z^2 - q_x^2 - q_y^2 \end{pmatrix}. \tag{2.1.2}$$

A translation vector may be defined with a scalar valued vector $q_T = [q_4\ q_5\ q_6]$, then the complete transformation vector is $q = [q_R|q_T]^t$. After rotation and translation of the shape P, the mean square objective function between P and P' becomes:

$$f(q) = \frac{1}{N_P} \sum_{i=1}^{N_p} \|p'_i - R(q_R)p_i - T(q_T)\|^2, \tag{2.1.3}$$

To obtain the transformation vector, we need to find the main dimension in the energy matrix, which is derived from the cross-covariance matrix of the two point sets P and P'. The cross-covariance-matrix $\sum_{PP'}$ is computed as:

$$\sum_{PP'} = \frac{1}{N_P} \sum_{i=1}^{N_p} [(p_i - \mu_P)(p'_i - \mu'_P)^t],$$

Input: P,X

Output: q_{N_0},d_{N_0}

repeat:

1. Find the closest points from X to all the points p_i of P, i.e. build the closest points set Y_k with $Y_k = C(P_k, X)$, where C is the closest points operator.
2. Apply the least-squares registration technique of equation 2.1.4 to P_0 and Y_k as $(q_k, d_k) = Q(P_0, Y_k)$.
3. Update P as $P_{k+1} = q_k(P_0)$, i.e. apply the transformation to P_0.

until:
maximal number of iterations N_{iter} has reached or $d_k - d_{k-1}$ is smaller than a threshold.

Figure 2.1: The ICP-Algorithm.

where $\mu_P = \frac{1}{N_P}\sum_{i=1}^{N_P} p_i$ and $\mu'_P = \frac{1}{N'_P}\sum_{i=1}^{N'_P} p'_i$ is the center of mass of the point cloud P and P', which contains N_P and N'_P points respectively.

With this cross-covariance-matrix a 4×4 symmetric energy matrix $Q(\sum_{PP'})$ is formed in the following way:

$$Q(\sum_{PP'}) = \begin{pmatrix} tr(\sum_{PP'}) & \Delta^t \\ \Delta & \sum_{PP'} + \sum_{PP'}^t - tr(\sum_{PP'})I_3 \end{pmatrix},$$

where $\Delta = [a_{23}\ a_{31}\ a_{12}]$ and $a_{ij} = (\sum_{PP'} - \sum_{PP'}^t)_{ij}$. I_3 is a 3×3 identity matrix. The unit eigenvector $q_R = [q_0\ q_x\ q_y\ q_z]$ corresponding to the largest eigenvalue of the matrix $Q(\sum_{PP'})$ gives us the optimal rotation. This is actually the unit quaternion to find the rotation matrix R by using equation (2.1.2). The optimal translation is then found by:

$$q_T = \mu_{P'} - R(q_R)\mu_P.$$

In the rest of this thesis, all these least-squares quaternion optimization for the state registration vector will be denoted as:

$$(q, d_{ms}) = Q(P, X), \tag{2.1.4}$$

where q is the transformation vector and d_{ms} is the mean-square error of the optimization function in equation (2.1.3). For the third step, the transformation will be

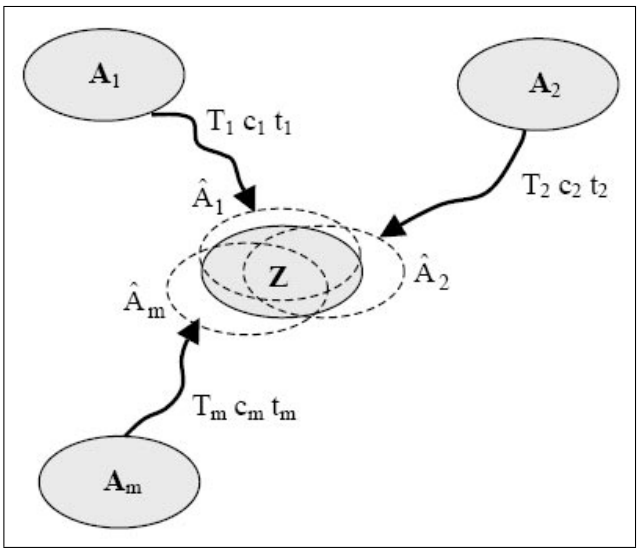


Figure 2.2: Generalized Procrustes analysis.

applied to all the points in point set P. Afterwards, the position of P will be updated through $P = q_k(P_0) = R(q_R)P_0 + q_T$.

These three steps will be repeated until maximal number of iterations have been reached or the current value of error d_{ms} can be no longer minimized.

There are also extensions to the standard ICP algorithm. Usually these variations of ICP are more efficient and robust than the standard one. Since the ICP algorithm is a non-linear searching algorithm, problems of non-linear optimization also exist in this case: although the error function converges monotonously to a local minimum, it may be not a global minimum. And also the speed of convergence depends on the used data set.

2.1.2 Procrustes Analysis

In some statistical shape distribution analysis, a mean shape is required to be computed from several data shapes. However, these shapes may have different size, rotation angle and position. We can not simply average all these shapes to get the mean shape. The generalized Procrustes analysis proposed by Gower in [20] is an algorithm to align a set of shapes in a least-squares sense to their mutual mean by estimating the transform parameters. As shown in Fig. 2.2 (taken from [2]), different shapes are transformed to their mean and common coordinate system.

In the Procrustes analysis, the objective function to minimize is:

$$min\ tr\left\{\sum_{i=1}^{m}\sum_{j=i+1}^{m}[(c_iA_iT_i+jt_i^T)-(c_jA_jT_j+jt_j^T)]^T[(c_iA_iT_i+jt_i^T)-(c_jA_jT_j+jt_j^T)]\right\}, \tag{2.1.5}$$

where $\mathbf{tr}\{\}$ stands for trace of the matrix, $A_1, A_2, \ldots, A_m$ are m model point matrices, which contain the same set of p points in k dimensional m different coordinate systems, c_i is the scaling factor of model A_i, T_i is the orthogonal transformation matrix of A_i, t_i is the translation vector of A_i and j is a unit vector. The optimization function is then the trace of a cross-covariance matrix. According to Goodall (1991), there is a matrix Z, also named consensus matrix, that contains the true coordinates of the p points defined in a mean and common coordinate system (Fig. 2.2). The solution of the problem can be thought as the search of the unknown optimal matrix Z.

$$Z + E_i = \hat{A}_i = c_iA_iT_i + jt_i^T,\ i\ =\ 1, 2, \ldots, m$$

$$vec(E_i) \sim N\left\{0, \sum = \sigma^2(Q_P \otimes Q_K)\right\}$$

where E_i is the random error matrix in normal distribution, $\sum$ is the covariance matrix, Q_P is the cofactor matrix of the p points, Q_K is the cofactor matrix of the k coordinates of each point, $\otimes$ stands for the Kronecker product, and σ^2 is the variance factor.

Least squares estimation of unknown transformation parameters T_i, c_i, and t_i ($i = 1, 2, \ldots, m$) must satisfy the following objective function, as mentioned before in equation (2.1.5).

$$\min\left\{\sum_{i=1}^{m}\sum_{j=i+1}^{m}\|\hat{A}_i - \hat{A}_j\|^2\right\} \tag{2.1.6}$$

Let us define a matrix C that is the geometrical centroid of the transformed matrices as follows:

$$C = \frac{1}{m}\sum_{i=1}^{m}\hat{A}_i. \tag{2.1.7}$$

Then we can transform the equation (2.1.6) into the following form with the use of matrix C:

$$\sum_{i=1}^{m}\sum_{j=i+1}^{m}\|\hat{A}_i - \hat{A}_j\|^2 = m\sum_{i=1}^{m}\|\hat{A}_i - C\|^2 = m\sum_{i=1}^{m}tr\left\{(\hat{A}_i - C)^2(\hat{A}_i - c)\right\} \tag{2.1.8}$$

The solution of the generalized Procrustes problem can be achieved by minimizing the right side of equation (2.1.8) in an iterative computation scheme. The pseudo-code in Fig. 2.3 indicates the main steps of the generalized Procrustes analysis.

This procedure aligns all the shapes $\{A_i\}$ onto their means. However, a procedure for the solution to the first step of each iteration in the above algorithm, i.e. aligning

initialize:
Define the initial centroid.

repeat:

1. Compute the similarity transformation parameters T_i, c_i, and t_i for each model points matrix A_i with respect to the centroid C.
2. After the calculation of each matrix $\hat{A}_i$ is carried out, iterative updating of the centroid C according to the equation (2.1.7).

until:
Global convergence, i.e. stabilization of centroid C.

Figure 2.3: Algorithm for the Generalized Procrustes analysis

two shapes, is still required. The solution is obtained with extended orthogonal Procrustes analysis. Let the matrix S be:

$$S = A^T(I - \frac{jj^T}{p})C.$$

Then the solution to the orthogonal transformation matrix T is found by singular value decomposition (SVD) as:

$$S = VDW^T$$
$$T = VW^T$$

where the matrix V is the orthonormal eigenvector matrix of SS^T, the matrix W is the orthonormal eigenvector matrix of S^TS, and the diagonal of matrix D is the singular values of the matrix S. The equation for the scaling factor c is:

$$c = \frac{tr\left\{T^TA^T(I - \frac{jj^T}{p})C\right\}}{tr\left\{A^T(I - \frac{jj^T}{p})A\right\}}$$

Finally, translation vector t can be solved as:

$$c = \frac{(C - cAT)^Tj}{p},$$

The detailed derivations of these equations are given in [20].

When these equations are plugged into the first step of the generalized Procrustes iteration and the transformations are applied to A_i, it becomes possible to align n different shapes on their mutual means.

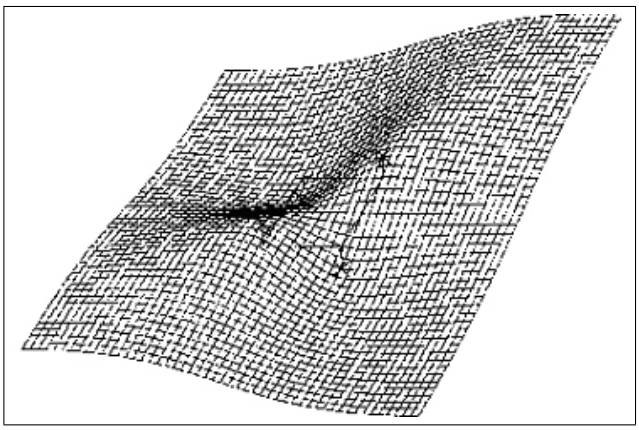


Figure 2.4: Deformation of a thin plate (taken from [5])

2.1.3 Thin Plate Spline (TPS) Warping

In contrast to the rigid transformation in ICP, the thin plate spline (TPS) algorithm offers a non-linear deformation that maps a shape onto another shape. The thin plate spline is a commonly used basis function for representing coordinate mappings from $\mathbb{R}^2(\mathbb{R}^3)$ to $\mathbb{R}^2(\mathbb{R}^3)$. Bookstein [5], for example, has studied its application to the problem of modeling changes in biological forms. In its regularized form the TPS model includes the affine model as a special case.

The name "thin plate spline" refers to a physical analogy of thin metal sheet bending in such a way, that the points on surface are forced to move from original position to the specified target position. An example to this type of deformation is shown in Fig. 2.4. With the use of TPS warping, one can deform a shape by transforming its landmark points onto another shape's landmark points in such a way that the total bending energy of all points is minimized.

Let v_i denote the target function values at locations (x_i, y_i) in the x-y plane, with $i = 1, 2, \ldots, p$. In particular, we will set v_i equal to the target coordinates (x'_i, y'_i) in turn to obtain one continuous transformation for each coordinate. We assume that the locations (x_i, y_i) are all different and are not collinear. The TPS interpolant $f(x, y)$ minimizes the bending energy:

$$I_f = \iint_{R^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)\,\mathrm{d}x\,\mathrm{d}y,$$

and has the form

$$f(x, y) = a_1 + a_x x + a_y y + \sum_{i=1}^{p} w_i U\big(\big\|(x_i, y_i) - (x, y)\big\|\big),$$

where $U(r) = r^2 \log r^2$ is the basis function of TPS interpolation, r is the distance $\sqrt{x^2 + y^2}$ from the Cartesian origin. The coefficient w_i is the weight of the basis

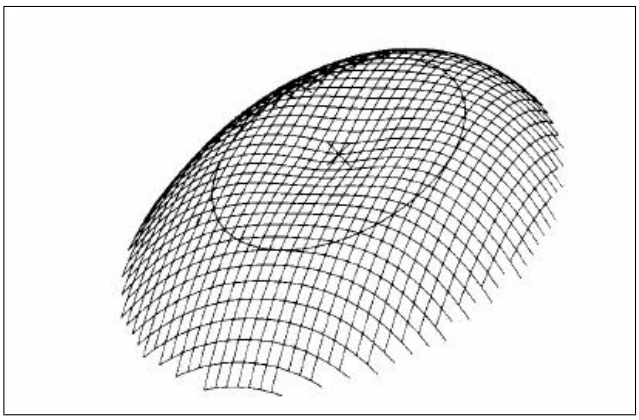


Figure 2.5: Shape of a thin plate lofted at the origin of Cartesian coordinate system (taken from [5])

function on the point of (x_i, y_i). This basis function as sketched in Fig. 2.5 is the fundamental solution to the following bi-harmonic equation.

$$\Delta^2 U = \Big(\frac{\partial^2}{\partial_{x^2}} + \frac{\partial^2}{\partial_{y^2}}\Big)^2 U = 0$$

In order for $f(x_i, y_i)$ to have square integrable second derivatives, we require that

$$\sum_{i=1}^{p} w_i = 0 \qquad and \qquad \sum_{i=1}^{p} w_i x_i = \sum_{i=1}^{p} w_i y_i = 0.$$

Together with the interpolation conditions, $f(x_i, y_i) = v_i$, a linear system for the TPS coefficients is yielded as:

$$\begin{pmatrix} K & P \\ P^T & O \end{pmatrix} \begin{pmatrix} w \\ a \end{pmatrix} = \begin{pmatrix} v \\ o \end{pmatrix},$$

where

$$K = \begin{pmatrix} 0 & U(r_{12}) & \dots & U(r_{1p}) \\ U(r_{21}) & 0 & \dots & U(r_{2p}) \\ \dots & \dots & \dots & \dots \\ U(r_{p1}) & U(r_{p2}) & \dots & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \dots & \dots & \dots & \dots \\ 1 & x_p & y_p & z_p \end{pmatrix}$$

here K is a kernel coefficients matrix, and P is a $p \times 4$ source landmarks' homogenous coordinates matrix. O is a 4×4 matrix of zeros, o is a 4×1 column vector of zeros, w and v are column vectors formed from w_i and v_i, respectively, and a is the column vector with elements a_1, a_x, a_y. We will denote the $(p+4) \times (p+4)$ matrix of this system by L, then the transformation matrix that maps the source landmarks to the target landmarks becomes:

$$(w \mid a_1 a_x a_y)^T = L^{-1}(v \mid o)^T,$$

The deformation is then interpolated by a linear combination of the basis function around the landmarks.

The following properties hold for this TPS solution:

1. $f(x_i, y_i) = v_i$, this is actually due to the equations in the first p rows of L.
2. The function f minimizes the quantity $I_f = \iint_{R^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2)\,\mathrm{d}x\,\mathrm{d}y$, I_f is called the integral bending norm.

2.2 Face Representation

Before face data is feed in a classification system, it should be represented in a more convenient manner. This feature extraction step, which directly affects the identification accuracy, is for representing the faces in such a way that faces of the same person are close to each other and faces of different people are further apart in this representation space. Here we first explain our local appearance-based approach in detail. We also compare our approach with several well-known face recognition algorithms: Eigenfaces [33], linear discriminant analysis (LDA) [37], Bayesian face recognition [27], embedded hidden Markov model (EHMM) [28], point set difference (PSD) [24] and point distribution model (PDM) [10]. An overview of these algorithms will be given in the following subsections.

2.2.1 DCT-based Local Appearance Approach

Appearance-based subspace approaches such as Eigenfaces have dominated the face recognition research during the last years. Experiments have shown that component based [21] and local model based approaches [29], which use local regions of salient features, are superior to the holistic template approaches. But, the detection of salient features -i.e. eyes- is not an easy task. Moreover, erroneous detection of these facial features may severely degenerate the performance. A local appearance based approach has been proposed in [19], which divides the input face image into non-overlapping blocks to perform Eigenfaces locally on each block. The experiments

conducted in [19] showed that the proposed method outperforms the standard holistic Eigenfaces approach under variations of expression and illumination.

In this work, we apply a novel local appearance based approach using discrete cosine transform (DCT). The underlying idea is to utilize local information while preserving the spatial relationships. In [16], the DCT is proposed to be used to represent the local regions. The DCT has been shown to be a better representation method for modeling the local facial appearance compared to principal component analysis (PCA) and the discrete wavelet transform (DWT) in terms of face recognition performance [16]. The discrete cosine transform for 2D input $f(x, y)$ is defined as:

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

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

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

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

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

The corresponding DCT bases are shown in Fig. 2.6. As can be seen from the top-left part of the basis functions, the $(0, 0)$ component represents the average intensity value of the depth image. From the figure, it can be also noticed that the $(0, 1)$ and $(1, 0)$ components represent the average vertical and horizontal changes, and the $(1, 1)$ component represents the average diagonal changes in the depth image. Lower order coefficients represent lower frequencies, whereas higher order coefficients correspond to higher frequencies.

Feature extraction from registered images using local appearance-based representation can be summarized as follows: The input depth image is divided into blocks of 8×8 pixels size. Each block is then represented by its DCT coefficients. These DCT coefficients are ordered using a zig-zag scanning pattern [17](see Fig. 2.7). From the ordered coefficients, M of them are selected according to the feature selection strategy resulting in an M-dimensional local feature vector. Finally, the DCT coefficients extracted from each block are concatenated to construct the overall feature vector of the corresponding image.

2.2.2 Eigenfaces

Eigenface is a well-known face recognition technique based on principal component analysis (PCA). PCA is the best, in the mean-square reconstruction error sense, linear

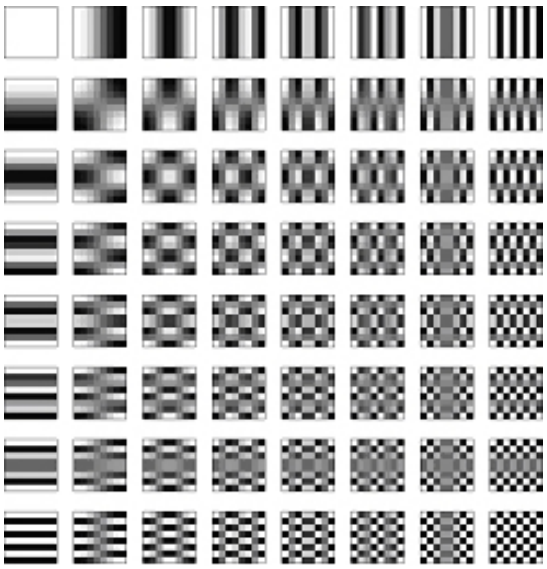


Figure 2.6: DCT basis functions for $N = 8$, The origin is at the top left corner.

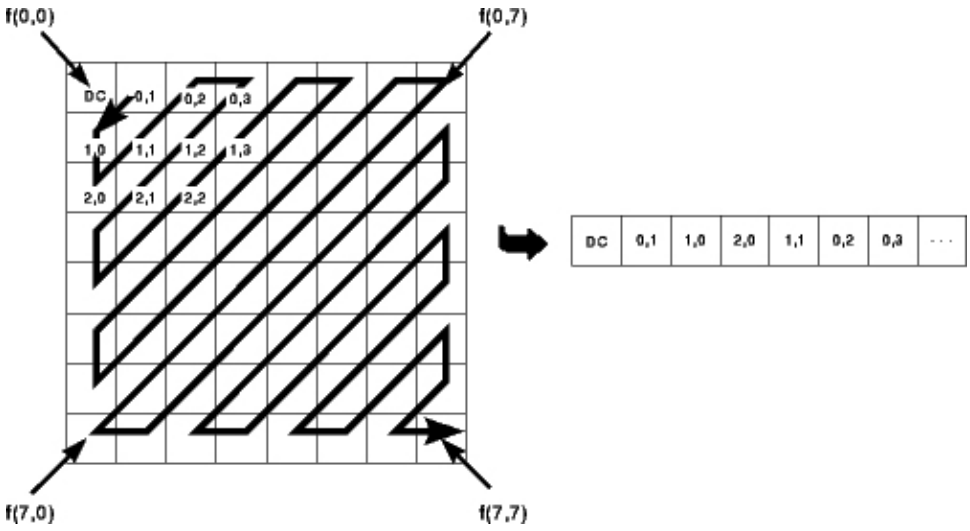


Figure 2.7: Zig-zag scanning, first coefficient is average intensity value of this block (DC).

dimension reduction technique. In various fields, it is also known as the Karhunen-Loève transform. In essence, PCA is used to reduce large dimensional data sets by finding a few orthogonal, uncorrelated principal components. The first principal component explains the largest percentage of the variation in the original high dimensional data set (and the second principal component explains the second largest percentage and so on). Typically the first few principal components account for most of the variation while the remaining PCs can be discarded with minimal loss of information.

In case of face recognition, we wish to find the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of the face images, treating an image as a vector in a very high dimensional space. The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images. An N by N face image can be represented as a vector, Γ_i, by ordering the pixel values column-wise or row-wise. Calculation of eigenfaces with PCA is detailed in following steps:

1. Prepare an initial set of face images $\{\Gamma_i\}$(the training set).
2. Calculated the average vector $\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$. Each face differs from the average by the vector $\Phi_i = \Gamma_i - \Psi$.
3. The covariance matrix C is calculated according to $C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T$.
4. Calculate the eigenvectors (eigenfaces), u_i, and eigenvalues, λ_i, of the covariance matrix. The eigenvectors (eigenfaces) must be normalized so that they are unit vectors, i.e. of length 1.
5. From the M eigenvectors (eigenfaces), u_i, only M' of them should be chosen, which correspond to the highest eigenvalues. Eigenvectors corresponding to large eigenvalues are the directions in which the data has strong component, or equivalently large variance. Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of characteristic features of the faces.

After M' eigenfaces, $u_i s$, are determined, the training phase of the algorithm is finished. The process of classification of a new (unknown) face Γ_{new} to one of the classes (known faces) proceeds in two steps. First, the new image is transformed into its eigenface components. The resulting weights form the weight vector $\Omega_{new}^T = [\omega_1\ \omega_2\ \ldots\ \omega_M]$, where $\omega_k = u_k^T(\Gamma_{new} - \Psi),\ k = 1..M$.

The Euclidean distance between two weight vectors $d(\Omega_i, \Omega_j)$ provides a measure of similarity between the corresponding images i and j. If the Euclidean distance between Γ_{new} and the other faces exceeds some threshold value θ, one can classify Γ_{new} as "unknown", and optionally use it to create a new face class.

2.2.3 Linear Discriminat Analysis (LDA)

Given a set of N images, where each image belongs to one of the c classes, LDA selects a linear transformation matrix W in such a way that the ratio of the between-class

scatter to the within-class scatter is maximized. Let the between-class scatter matrix be defined as

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T,$$

and the within-class scatter matrix be defined as

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T,$$

where μ_i denotes the mean image of class X_i and N_i denotes the number of images in class X_i. If S_W is nonsingular, LDA will find an orthonormal matrix W_{opt} maximizing the ratio of the determinant of the between-class scatter matrix to the determinant of the within-class scatter matrix. That is, the LDA projection matrix is represented by

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_i \ w_2 \ \dots \ w_m]. \tag{2.2.1}$$

The set of the solution $\{w_i \mid i = 1, 2, \dots, m\}$ is that of the generalized eigenvectors of $S_W^{-1} S_B$ corresponding to the m largest eigenvalues $\{\lambda_i \mid i = 1, 2, \dots, m\}$, i.e.,

$$S_B w_i = \lambda_i S_W w_i, \qquad i = 1, 2, \dots, m$$

$$(S_W^{-1} S_B) w_i = \lambda_i w_i$$

if S_W is a non-singular matrix then the optimation criterion described in equation (2.2.1) is maximized when the projection matrix W_{opt} is composed of the eigenvectors of $S_W^{-1} S_B$ with at most $(c - 1)$ nonzero corresponding eigenvalues, where c is the number of classes. This is the standard LDA procedure.

The performance of the standard LDA can be seriously degraded if there are only a limited number of total training observations N compared to the dimension of the feature space n. Since the rank of the within-class scatter matrix S_W is at most $(N - c)$, and in general, the number of images in the training set, N, is much smaller than the number of pixels in each image, n, S_W is a singular matrix, which is not invertible.

The Fisherfaces [3] method is one of the most successful feature extraction approaches for solving limited sample size problems in face recognition. This method is essentially a two-stage dimensionality reduction technique. First the face images from the original vector space are projected to a lower dimensional space using PCA and then LDA is applied next to find the best linear discriminant features on that PCA subspace.

More specifically, the Fisherface projection matrix W_{opt} can be calculated as

$$W_{opt} = W_{lda} * W_{pca},$$

where W_{pca} is the projection matrix from the original image space to the PCA subspace, and W_{lda} is the projection matrix from the PCA subspace to the LDA subspace obtained by maximizing the ratio

$$W_{lda} = arg \max_{P} \frac{\left|W^T W_{pca}^T S_B W_{pca} W\right|}{\left|W^T W_{pca}^T S_W W_{pca} W\right|}.$$

If $W_{pca}^T S_W W_{pca}$ is a non-singular matrix then the Fishers criterion is maximized when the projection matrix W_{lda} is composed of the eigenvectors of $(P_{pca}^T S_W P_{pca})^{-1}(P_{pca}^T S_B P_{pca})$ with at most $(c-1)$ nonzero corresponding eigenvalues.

The singularity problem of the within-class scatter matrix S_W is then overcome if the number of retained principal components varies from at least c to at most $(N-c)$ PCA features.

2.2.4 Embedded Hidden Markov Models (EHMM)

Hidden Markov models (HMM) have been successfully used for speech and activity recognition where data is one-dimensional. But for two-dimensional face recognition problem, using 2-D HMMs are too complex for real-time applications. The embedded HMM approach proposed in [28] considered the significant facial features of a frontal face image including the hair, forehead, eyes, nose and mouth as super observation vectors of a one dimensional HMM, which are also called super states. Because of these states themselves are HMMs that consist of several embedded states. The super states may then be used to model two-dimensional data along one direction, with the embedded HMM modeling the data along the other direction. This model differs from a true two-dimensional HMM since transitions between the states in different super states are not allowed. Therefore, this is approach is named as an embedded HMM.

Although an embedded HMM is more complex than a one-dimensional HMM, it is more appropriate for two-dimensional data. The salient facial regions of a face can be model as the states of an embedded HMM. Each state in the overall top-to-bottom HMM is assigned to a left-to-right HMM. This model is appropriate for face images since it exploits an important facial characteristic: frontal faces preserve the same structure of " super states" from top to bottom, and also the same left-to-right structure of "states" inside each of these "super states".

The observation sequence is generated using a window scanning the image from left to right, and top to bottom. A 2D-DCT is taken on each scanned image block, and the corresponding DCT coefficients are considered as an observation vector. The compression and decorrelation properties of the 2D-DCT for images make it suitable their use as observation vectors. As described in our DCT based local appearance approach, the lower frequency DCT coefficients, which represent the most of the image energy, are used as observation vectors. And due to the compact representation power of DCT, the size of the observation vectors is dramatically reduced, which decreases the computational complexity of the algorithm.

2.2.5 Bayesian Face Recognition

The Bayesian face recognition algorithm is a probabilistic similarity measure for image matching based on a Bayesian analysis. Face images in training set are modeled with two classes of variation in appearance: intra-personal and extra-personal. The probability density functions for each class are then estimated from training data and used to compute a similarity measure based on the *a posteriori* probabilities.

The standard template-matching approaches to recognition often make use of simple image similarity metrics such as Euclidean distance or normalized correlation. For example, the similarity measure $S(I_1, I_2)$ between two images I_1 and I_2 can be set to be inversely proportional to the norm $\|I_2 - I_1\|$. This simple formulation requires precise alignment of the faces in the image, and it does not exploit knowledge of which type of variations are critical in expressing similarity. With the approach proposed in [27], the image-based differences, denoted by $\Delta = I_1 - I_2$, are extracted to characterize typical variations between two facial images. The authors defined two distinct and mutually exclusive classes: Ω_I representing *intrapersonal* variations, the variations between multiple images of the same individual (e.g., with different expressions and lighting conditions), and Ω_E representing *extrapersonal* variations, the variations between images of different individuals. The similarity measure is then expressed in terms of the probability

$$S(I_1, I_2) = P(\Delta \in \Omega_I) = P(\Omega_I \mid \Delta) = \frac{P(\Delta|\Omega_I)P(\Omega_I)}{P(\Delta|\Omega_I)P(\Omega_I) + P(\Delta|\Omega_E)P(\Omega_E)}$$

where $P(\Omega_I \mid \Delta)$ is the *a posteriori* probability given by Bayes rule, using estimates of the likelihood $P(\Delta \mid \Omega_I)$ and $P(\Delta \mid \Omega_E)$ which are derived from training data using an efficient subspace method for density estimation of high-dimensional data. This particular Bayesian formulation transforms the standard face recognition task into a binary pattern classification problem with Ω_I and Ω_E. The problem is then solved using the maximum *a posteriori* (MAP) rule: two images are determined to be the same individual if $P(\Omega_I|\Delta) > P(\Omega_E|\Delta)$, or if $S(I_1, I_2) > \frac{1}{2}$.

2.2.6 Point Set Distance (PSD)

Point set distance (PSD) is a simple 3D face recognition algorithm based on point cloud volume difference. This algorithm requires a point-to-point correspondence among all faces. This correspondence is based on shape registration and mesh resampling technique, which will also be applied in our 3D face registration procedure. The total distance of the point sets, i.e. the volume difference between two faces is used as a similarity measure between two faces. A test face with the minimum volume difference is classified as the most similar person. The similarity measure between a test face X and a training face P can be calculated as:

$$Similarity(X, P) = -\sum_{i=1..N_M} \|x_i - p_i\|$$

If the minimum volume difference of a test face and a training face, i.e. the distance of a test person to its nearest neighbor, exceeds a threshold, the face is considered to be an unknown person.

2.2.7 Point Distribution Model (PDM)

Point distribution model (PDM) is a statistical shape model. The principle behind the PDM is that the shape and deformation of an object can be expressed statistically by formulating the shape as a vector representing a set of points that describe the object. This shape and its deformation (expressed with a training set, indicative of the object deformation) can then be learnt through statistical analysis. The training process is described in more detail below.

Given a collection of E training images of registered faces, the coordinates of N points maintain the same vertex index. A training sample e is represented by a vector $x_e = (x_1,\ y_1,\ z_1,\ \ldots, x_N,\ y_N,\ z_N)$. After aligning all the examples (using registration technique), the mean face shape $\bar{x}$ is calculated simply by finding the mean position of each landmark point.

In order to discover the major modes of variation, principal component analysis (PCA) is performed on the deviation of examples from the mean. For this, the covariance matrix S of the deviations must first be calculated:

$$S = \frac{1}{E}\sum_{e=1}^{E}(x_e - \bar{x})(x_e - \bar{x})^T$$

The t unit eigenvectors of S corresponding to the t largest eigenvalues supply the variation modes; t is generally much smaller than N, thus giving a very compact model. A face, x, is generated by adding linear combinations of these t eigenvectors, v_j, to the mean face shape:

$$x = \bar{x} + \sum_{j=1}^{t} b_j v_j,$$

where b_j is the weighting for the j^{th} eigenvector. Suitable limits for b_j are $\pm 3\sqrt{\lambda_j}$, where λ_j is the j^{th} largest eigenvalue of S. The face shape is given entirely in terms of these weights b_j, hence very few parameters are required to specify a face's exact position and shape. Fig. 4.2 illustrates the average face and some synthesized faces with different number of principal components. The faces reconstructed with more PCs are more similar to the original face.

2.3 Classification

After the features are extracted from a test face in the face representation phase, it is identified by comparing it with features from predefined classes stored in the

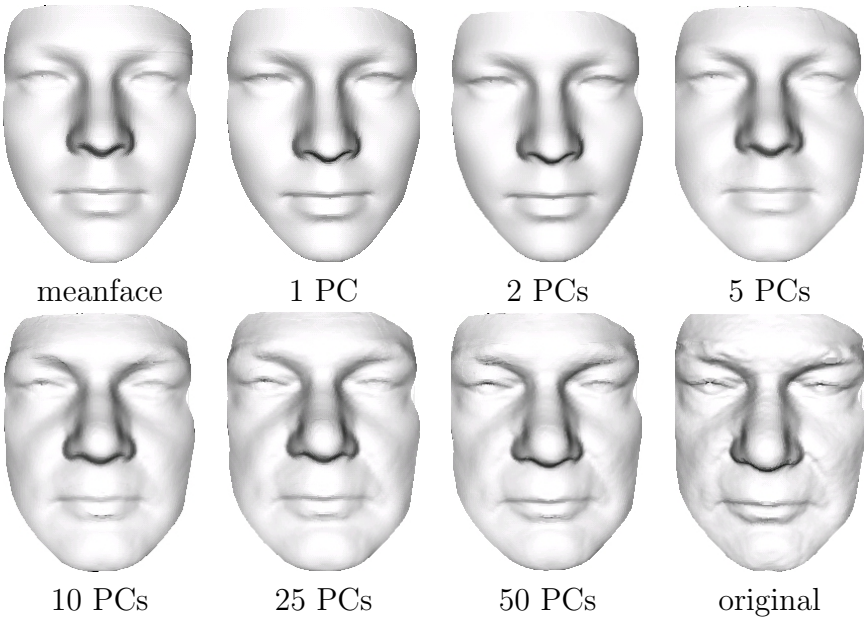


Figure 2.8: 3D face reconstructed with different number of PCs.

database.

2.3.1 Nearest Neighbor

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

Usually, the Euclidean distance metric is used to measure the nearest k neighbors in Euclidian space. Given the training set $T = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$, to determine the class of the input vector x. The most special case is the 1-NN method as show in Fig. 2.9, which just searches the nearest neighbor:

$$j = \arg \min_i \|x - x_i\|,$$

then, (x, y_j) is the solution. Euclidean distance is a distance metric based on L2 norm, so it is also called L2 norm distance. Other norms are also considered as distance norms, such as L1 norm as defined follows:

$$\text{L1 norm:} \qquad d = \sum_{m=1}^{M} \left| f_{training,m} - f_{test,m} \right|,$$

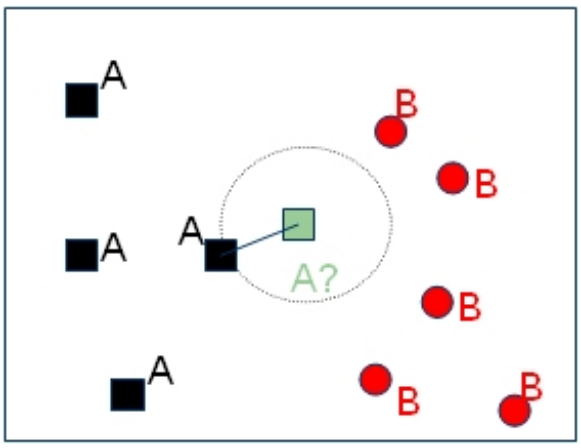


Figure 2.9: Nearest neighbor.

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

The distance metrics based on correlation and covariance are also discussed in [14] for nearest neighbor classifier. And a detailed performance comparison between these distance metrics with our local appearance face recognition algorithm is also given in [14].

2.3.2 Support Vector Machines (SVMs)

Support vector machines (SVMs) are supervised learning methods used for classification and regression. With the kernel trick, we can apply linear classification techniques to non-linear classification problems. In this work, a multi-class classifier is needed to classify one face class against all other face classes. In the following, we first explain the basics of the SVM for binary classification. Then we discuss how this technique can be extended to deal with multi-class classification cases.

Binary Classification

SVM belongs to the class of maximum margin classifiers. They perform pattern recognition between two classes by finding a decision surface that has maximum distance to the closest points in the training set which are called support vectors. Assume we have a training set of points $x_i \in \mathbb{R}^n, i = 1, 2, \ldots, N$, where each point belongs to one of the two classes identified by the label $y_i \in \{-1, 1\}$. Assuming linearly separable data, the goal of maximum margin classification is to separate the two classes by a hyperplane such that the distance to the support vectors is

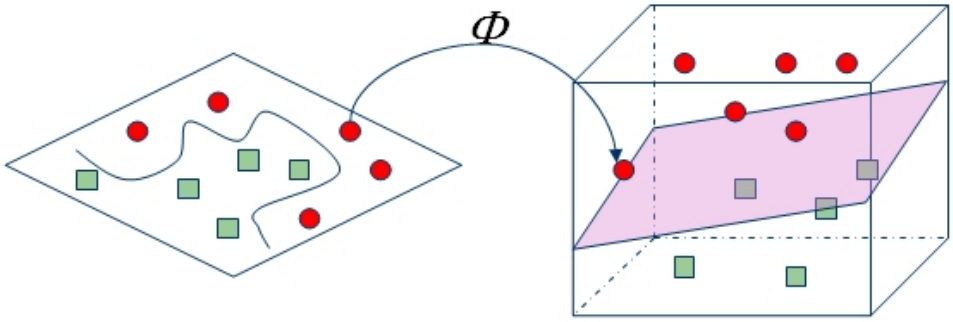


Figure 2.10: Support Vector Machine.

maximized. This hyperplane has the form:

$$f(x) = \sum_{i=1}^{l} a_i y_i x_i \cdot x + b_i. \tag{2.3.1}$$

The coefficients a_i and the b_i in equation (2.3.1) are the solutions of a quadratic programming problem. Classification of a new data point x is performed by computing the sign of the right side of equation (2.3.1). In the following we will use

$$d(x) = \frac{\sum_{i=1}^{l} a_i y_i x_i \cdot x + b}{\| \sum_{i=1}^{l} a_i y_i x_i \|}$$

to perform multi-class classification. The sign of d is the classification result for x, and $|d|$ is the distance from x to the hyperplane. The larger is the distance $|d|$, the more reliable the classification result. The entire construction can be extended to the case of nonlinear separating surfaces. Each point x in the input space is mapped to a point $z = \Phi(x)$ of a higher dimensional space (see Fig. 2.10), called the feature space, where the data are separated by a hyperplane. The key property in this construction is that the dot product of the two points in the feature space $\Phi(x) \cdot \Phi(y)$ can be rewritten as a kernel function $K(x, y)$. The decision surface has the equation:

$$f(x) = \sum_{i=1}^{l} y_i a_i K(x, x_i) + b_i.$$

Again, the coefficients a_i and b_i are the solutions of a quadratic programming problem. Note that $f(x)$ does not depend on the dimensionality of the feature space.

The most popular and powerful kernel function is the Gaussian RBF (Radial Basis Function) kernel:

$$K(x, y) = e^{\frac{\|x-y\|^2}{\sigma^2}}$$

It is a powerful kernel as its effect is to create a small classification "hyperball" around an instance. Here σ is a measure of the radius of the "hyperball" around an instance.

Multi-class classification

There are two basic strategies for solving q-class problems with SVMs:

1. In the one-vs-all approach q SVMs are trained. Each of the SVMs separates a single class from all remaining classes.

2. In the pairwise approach $\frac{q(q-1)}{2}$ machines are trained. Each SVM separates a pair of classes. The pairwise classifiers are arranged in trees, where each tree node represents an SVM.

Regarding the training effort, the one-vs-all approach is preferable since only q SVMs have to be trained compared to SVMs in the pairwise approach. Since the number of classes in face recognition can be rather large we choose the one-vs-all strategy where the number of SVMs is linearly proportional with the number of classes.

Chapter 3

Local Appearance-based 3D Face Recognition

The local appearance-based 3D face recognition is described in following sections.

3.1 Registration

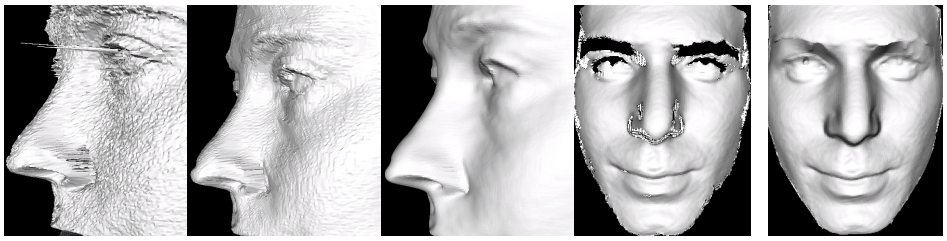
Recorded point clouds may have different poses and expressions. In order to extract proper local information from corresponding local facial blocks, a precise point-to-point correspondence should be established. In the following sections the processing steps of the face registration and depth image generation are explained.

3.1.1 Preprocessing

Range data acquired by 3D sensors may be noisy and sometimes may have spikes with sharp disparity discontinuities on the surface. To remove spike artifacts, we applied a median filter. In Fig. 3.1(a-b) the input noisy range image and the output of the median filtering are shown respectively. Afterwards, we used a Gaussian filter to make the face surface smoother as can be seen in Fig. 3.1(c). 3D laser scanners may sometimes have difficulties imaging the wet surfaces, such as eyeballs, and hairy surfaces, such as eyebrows, which may cause holes on those surfaces as illustrated in Fig. 3.1(d). These holes were filled using linear interpolation as shown in Fig. 3.1(e).

3.1.2 Dense Correspondence

To establish a dense correspondence between faces in training and testing sets, we transformed all faces to a common coordinate framework. This transform is based on the landmarks that are placed on the salient facial feature points. We first selected



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

Figure 3.1: (a) Original range image containing spikes and Gaussian noise. (b) Spikes are removed with a median filter. (c) Range image is smoothed with a Gaussian filter. (d) Holes on wet and hairy surface. (e) Hole-filling with linear interpolation.

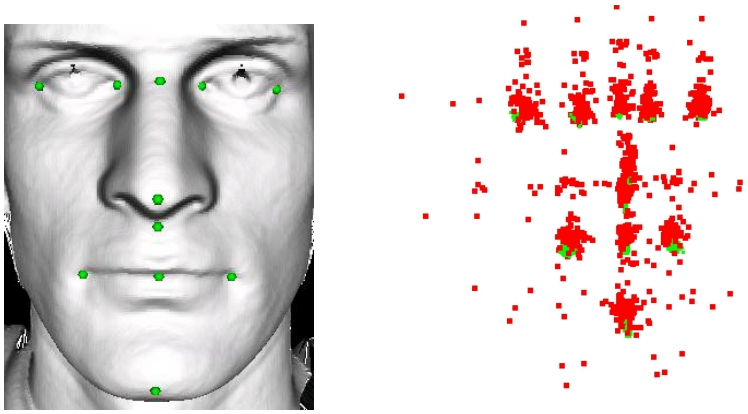
a face with the least number of points in the training set as a base mesh. Then, we placed 11 landmarks on all faces as shown in Fig. 3.2(a).

We can use any set of landmarks as the common frame of reference, but in order to apply statistical shape analysis such as point distribution model (PDM) [10], we used the generalized Procrustes algorithm to compute the mean landmarks. Each face is then warped onto the mean landmarks using the thin plate spline transform. Fig. 3.3 illustrates this process.

After aligning all the surfaces, a point-to-point dense correspondence was established by taking the surface's closest points to the vertices in the base mesh. Some faces in the data set may contain neck and ear while others may not, and since we are only interested in face region, the parts outside the face region should be removed. We discarded the vertices of the base mesh that have more than 20mm distance to the surfaces; for details see [23]. The remaining vertices then construct the final base mesh as shown in Fig. 3.4. Aligned faces that are sampled with this base mesh will contain the same number of vertices and same portion of the face region.

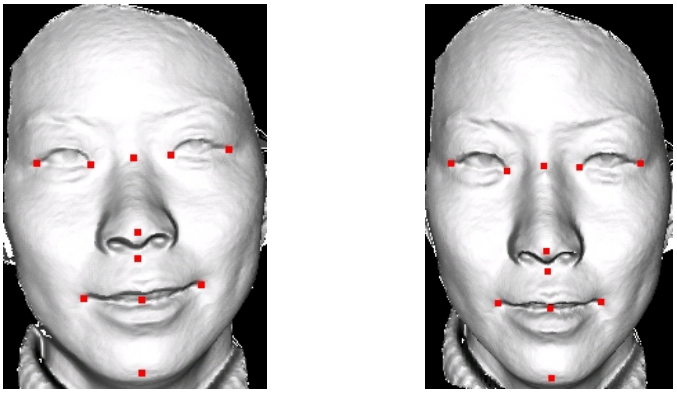
3.1.3 Depth Image Generation

Resampling with closest-point mapping may result in folds and uneven sampling in the surface where the correspondence between surfaces with high curvature is not very close. In such case, as shown in Fig. 3.5(a), the tip of target surface is not sampled at all, while the vicinity of that tip may be sampled twice, which introduces a fold into the final mesh. An example of a depth image generated from such mesh is shown in Fig. 3.5(c). Pixels around the nose do not correspond to the exact depth value on the original face, which would degrade the recognition performance. Since the base mesh and the target mesh are closely aligned, we used another resampling method illustrated in Fig. 3.5(b). Through each vertex on the base mesh, we cast a ray along z axis on the target surface. The resampled point is the crossing point if



(a) (b)

Figure 3.2: (a) Eleven landmarks are manually placed on the face surface: Four eye corners, saddle point at the top of the nose, nose tip, point under the nose over lip, two mouth corners, middle of the mouth and middle of the chin. (b) Red point sets are landmarks on all training faces, green point sets are landmarks aligned with generalized Procrustes algorithm.



(a) (b)

Figure 3.3: (a) Face before TPS warping. (b) Face after warping onto mean landmarks.

Figure 3.4: Base mesh selected from training set, cluster vertices are discarded with a distance threshold.

it exists. However, vertices at the border of the base mesh may sometimes have no corresponding point. Then the closest-point mapping was applied because curvatures at the face mesh border are usually low. Depth map images generated with ray-casting mapping are better in appearance as shown in Fig. 3.5(d).

After resampling, the z-value of each vertex on the reconstructed face can be considered as an intensity value in the corresponding depth image. Fig. 3.6 illustrates the overall process of face registration and depth image generation from a 3D image.

3.2 Automatic Registration

To construct a fully automatic recognition system, we need to register the face shapes automatically without interrupting the user. Iterative closest point (ICP) based registration is one of the most popular approaches for this purpose. However, ICP falls easily into local minima if two shapes are not coarsely matched (see Fig. 3.7). Therefore, an initial alignment was performed by matching nose tips on two faces. The nose tip on each face was estimated according to the depth values and geometric information around the nose.

ICP based approaches treat the face as a rigid object, which is not suitable for handling expression variations. Using a non-rigid deformation such as the thin-plate spline (TPS) technique can deform the faces with non-neutral expressions. Experiments in [26] show a substantial reduction of recognition error after the non-rigid deformation. To be able to perform TPS based automatic registration, we need to detect the landmarks automatically. The automatic landmark detection algorithm we used is a modified version of the one in [24] and is described as follows:

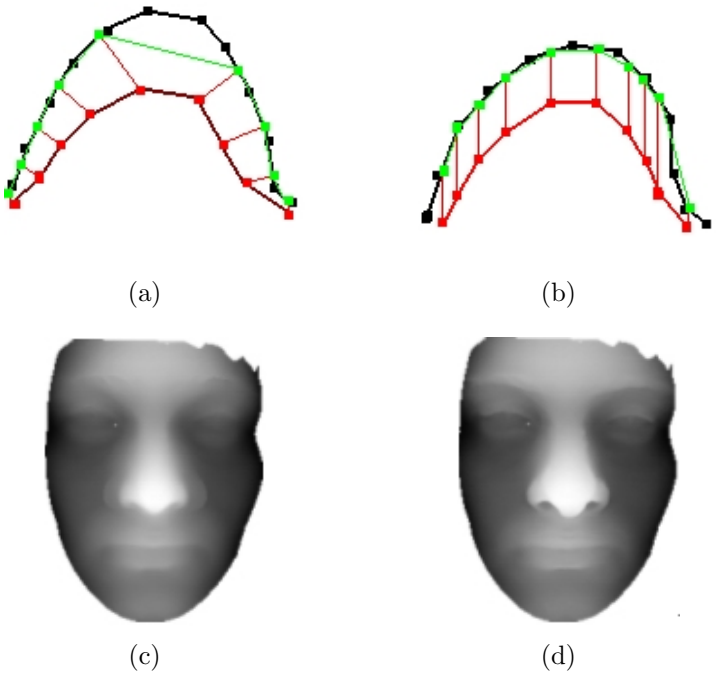


Figure 3.5: (a) Closest-point mapping, find the closest point on target mesh for each base mesh vertex. (b) Ray-casting mapping, find the crossing point on target mesh for each ray casting from base mesh vertices. (c) Depth map constructed with closest-point mapping. (d) Depth map constructed with ray-casting mapping.

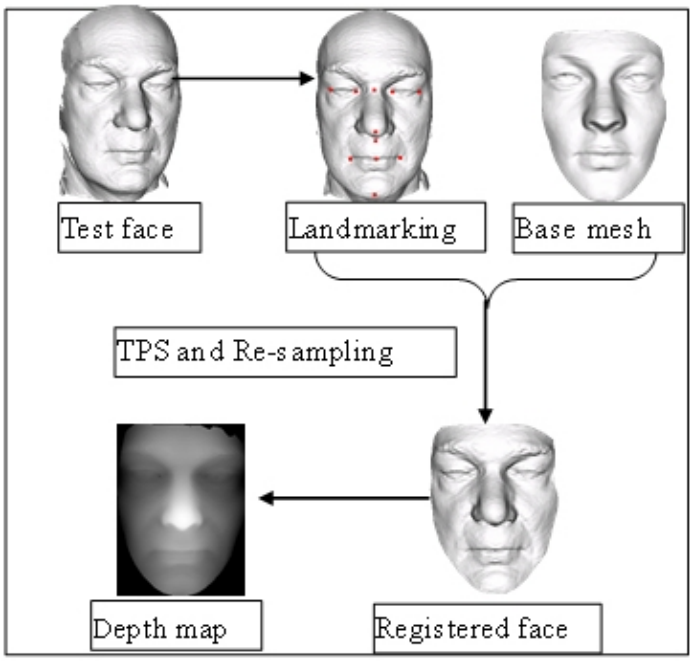


Figure 3.6: Face registration and depth map image generation.

Figure 3.7: ICP mismatch due to coarse alignment error.

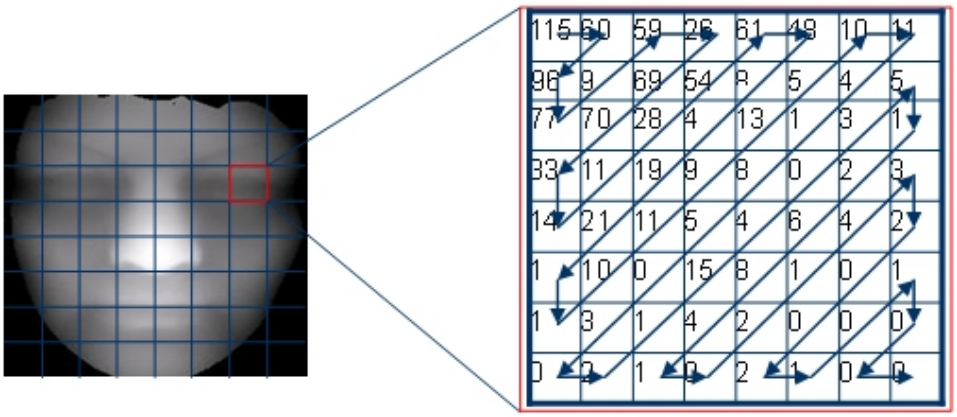


Figure 3.8: Local DCT feature vector from depth image.

1. Compute surface normals, Gaussian and mean curvatures for all the vertices in the test face surface.
2. Estimate the nose tip on the test face according to the depth value and geometric information,match the test face coarsely according to the estimated nose tip.
3. Perform ICP to align test face to the base mesh.
4. Estimate the position of initial landmarks on test face according to the landmarks on the base mesh.
5. Compute the symmetry plane based on the symmetry plane of the base mesh and the initial estimation of the landmarks.
6. Fine tune the initial landmarks according to their surface normals, curvatures and relative distance to symmetry plane.

After automatic landmarking, a point-to-point correspondence can be established with TPS deformation.

3.3 Feature Extraction

In the local appearance-based face representation approach, a depth image, which is generated by a registered and resampled 3D range image, is divided into blocks of 8x8 pixels size. On each 8×8 pixels block, a DCT is performed. The obtained DCT coefficients are ordered using zig-zag scanning as described in the methodology part (see Fig. 3.8).

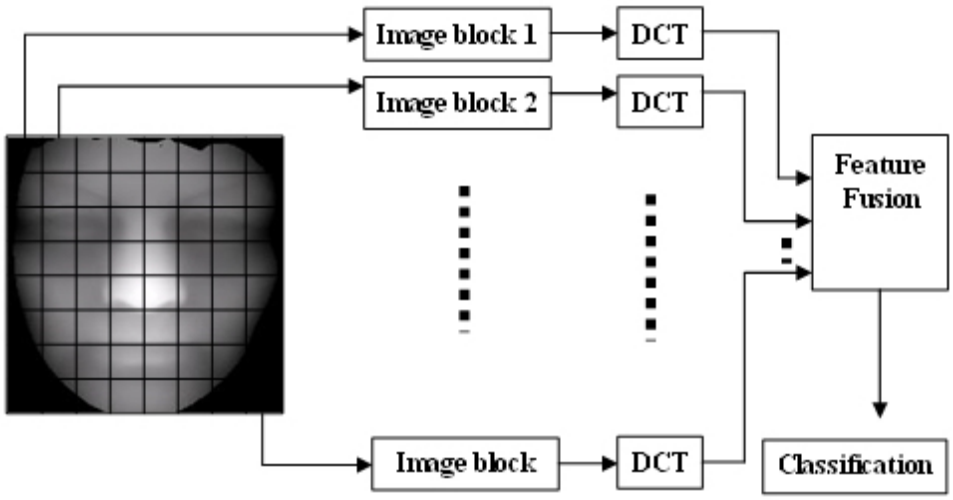


Figure 3.9: Local feature combination at the feature level.

From the ordered coefficients, according to the feature selection strategy, M of them are selected resulting an M dimensional local feature vector. Finally, the DCT coefficients extracted from each block are concatenated to construct the feature vector (Fig. 3.9).

3.4 Feature Selection and Feature Normalization

Feature selection and feature normalization is specifically important for a DCT-based face recognition scheme. In this section, we introduce the used feature sets and the normalization techniques that we investigated.

3.4.1 Feature Selection

The first three DCT coefficients contain general information about the global statistics of the processed block of an image. While the first coefficient represents the average intensity value of the whole block, the second and third coefficients represent the average horizontal and vertical intensity changes in the image block, respectively. It is important to assess the contributions of these features to the recognition performance. Therefore, we investigated the use of two different feature sets for classification, where the local feature vectors are constructed by one of the following methods:

1. Selecting the first M DCT coefficients.

2. Removing the first N coefficient, and selecting the first M DCT coefficients from the remaining ones.

We named the first feature set as DCT-0, the second one as DCT-N, where we choose $N = 1..6$.

3.4.2 Feature Normalization

There are two aspects in feature normalization. The first aspect is the total magnitude of each block's DCT coefficients. Since DCT is an orthogonal transformation and conserves all the energy of the processed input block, the blocks with different brightness levels lead to DCT coefficients with different value levels. Because of this reason, we normalized the local feature vector's, f's, magnitude to unit norm:

$$f_n = f/\|f\|,$$

where f_n represents the normalized feature vector. Another normalization method was also studied in [14], which is based on standard deviations of the DCT coefficient vectors.

3.5 Classification

We considered the nearest neighbor and support vector machine as the classifier for the normalized DCT feature vectors. For nearest neighbor classifier the $L1$ norm is used as the distance metric, since it has been shown that the $L1$ norm provides better results than the $L2$ norm and normalized correlation. A test sample face is assigned to the class of the closest training sample face. We also tested a support vector machine (SVM) classifier as a non-linear classifier. We used the LibSVM for SVM implementation. This tool offers a grid searching software to obtain the optimal parameters for the kernel functions.

Chapter 4

Experiments

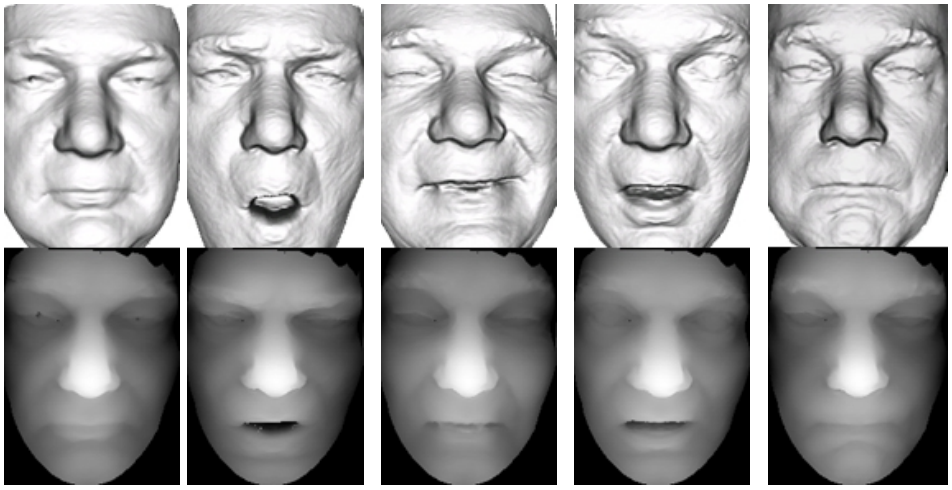
In this part, we first describe the face dataset we used. We analyze the results of the several experiments we have conducted. We also compare the recognition results of our local appearance based approach with other well-known face recognition approaches.

4.1 Experimental Setup

We conducted extensive experiments on the face recognition grand challenge (FRGC) version 2 data set to analyze the performance of the proposed local appearance based 3D face recognition approach. The 3D data corpus of FRGC database was collected by imaging subjects using a range scanner. We used the range images that were acquired in spring 2003 for training and the ones recorded in spring 2004 for testing. The training data contains neutral expressions, whereas the testing data contains different expressions, such as frowning, smiling, etc. In total we used 218 range images of 109 subjects for training where each individual has two samples, and 758 range images for testing where each individual has different number of samples ranging from two to twelve. Table 4.1 shows the data set configuration used in the experiments. Sample pre-processed range images and the corresponding registered depth images from the training and testing datasets are shown in Fig. 4.1. The depth images are scaled to 64×64 pixels resolution.

Training Set	Testing Set
109 Individuals	109 Individuals
218 Images	758 Images
2 images per subject	2-12 images per subject
Normal facial expression	Different facial expressions

Table 4.1: Data set used in the experiments.



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

Figure 4.1: First row: Pre-processed range images rendered with shade model in training and testing set. Second row: registered depth images. (a) neutral (b) frowning (c) smiling (d) surprised (e) puffy.

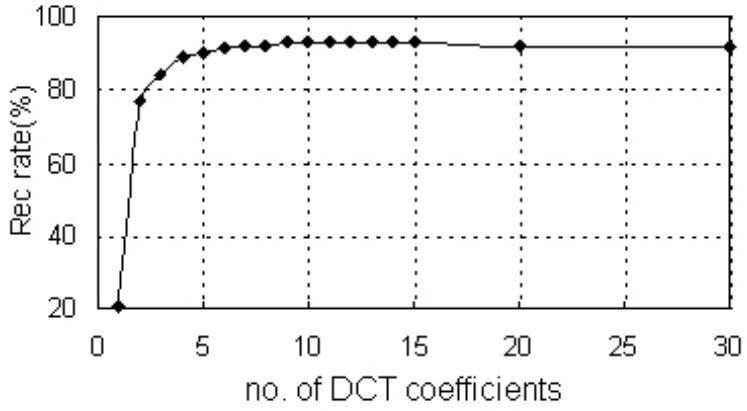


Figure 4.2: Correct recognition rate vs. local feature dimensionality.

4.2 Analysis of Local Appearance-based 3D Face Recognition

In the first part of the experiments, we analyzed the effects of local feature dimension, feature selection and face registration on the face recognition performance. Fig. 4.2 shows the face recognition performance with respect to increasing local feature dimensionality. In this experiment, the 3D faces were registered using all of the eleven, manually labeled landmark points, and depth images were generated using ray-casting. At each local block, the first coefficient was removed from the ordered DCT coefficients, since it only represents the average depth of a local image block. From the remaining coefficients the first M of them were selected. The selected local feature vector was normalized to have unit norm as suggested in [15] which has been shown to improve the face recognition performance. As can be observed from the figure, high correct recognition rates can be attained by using only five dimensional local feature vectors. The performance continues to increase slightly till to the feature dimension of ten. The correct recognition rate remains the same or decreases slightly, when the dimensionality increases further. Therefore, we choose to use ten dimensional local feature vectors for the rest of the experiments.

The second experiment assesses the effect of frequency content on face recognition performance. It is known that, lower order DCT coefficients represent the low frequency content of an image and have larger absolute values than those of higher order, high frequency DCT coefficients, which can be also observed from Fig. 3.8. Therefore, they have dominant effect in classification. In order to consider the correct recognition rate with different sets of features having different frequency contents, we discarded the first $N(N = 0..6)$ low frequency DCT coefficients and conducted the

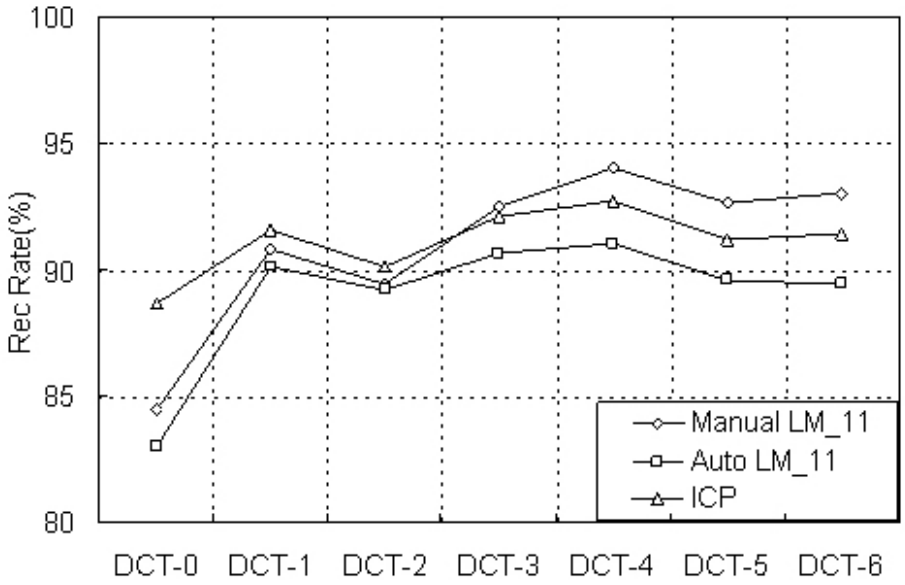
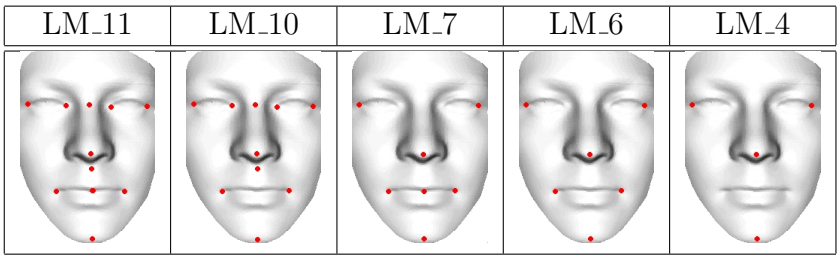


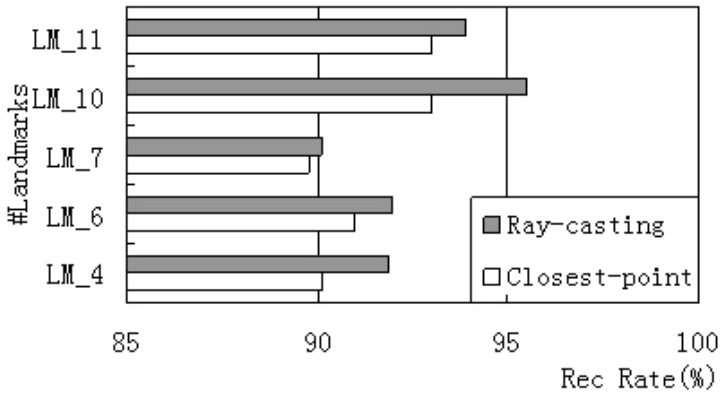
Figure 4.3: Recognition rate of DCT based local appearance approach using different feature sets methods. (DCT-N: Discard first N coefficients and select the first 10 coefficients from the remaining ones).

face recognition experiments. We ran the same experiments with three different registration setups to observe whether the selected features produce consistent results over each registration framework. The registration configurations were named "Manual LM_11", "Auto LM_11" and "ICP". "Manual LM_11" and "Auto LM_11" corresponded to face registration with 11 manually and automatically labeled landmarks, respectively, whereas "ICP" corresponded to registration with ICP. Ray-casting was used to generate depth images from the registered 3D faces. Correct recognition rates obtained from three different experimental setups are plotted in Fig. 4.3. In all of the experiments, the best results were obtained using the DCT-4 feature set, which implies that removing the coefficients that represent average horizontal and vertical changes as well as the one that represents the average depth, improves the face recognition performance.

The effects of used landmark points for registration and the depth image generation technique were analyzed in the third experiment. Usually more landmarks for registration may improve correspondence, but if the landmark points are poorly placed, correspondence may get worse. If more than necessary landmark points are used while performing the TPS warping, the cumulative noise of the landmarks may result in degenerate deformations. Therefore, we discarded some of the landmarks to analyze their effectiveness. In the experiment, five possible landmark combinations,



(a)



(b)

Figure 4.4: (a) Five landmarks combinations. (b) Recognition rate of DCT based local appearance approach with different landmark combinations.

illustrated in Fig. 4.4(a), were tested for registration. Both ray-casting and closest-point methods were used for depth image generation. The DCT-4 feature set was used for classification. The corresponding results can be seen in Fig. 4.4(b). We achieved the highest scores by selecting ten landmarks, excluding the landmark located in the middle of the mouth. This is expected, since this point is not easy to label on faces that have different facial expressions. As can be observed, ray-casting mapping always outperforms closest-point mapping. With optimal landmark combination and ray-casting, we achieved 95.5% correct recognition rate.

4.3 Recognition Based on Automatic Registration

We investigated the effect of automatic landmarking in the last experiment. The landmarks used for TPS warping are automatically detected. Fig. 4.5 displays the percentage of successful detection of some landmarks with respect to varying distance threshold. Table 4.2 shows the correct detection rate of all eleven landmarks. A

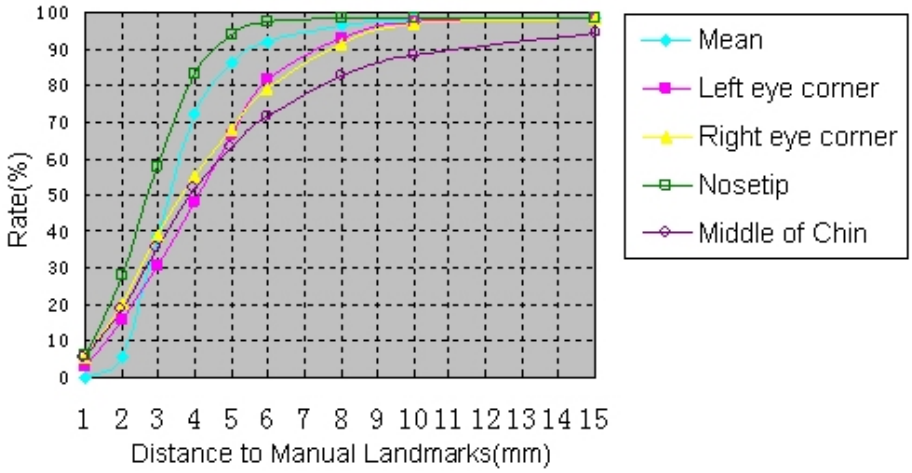


Figure 4.5: Correct detection rate of some landmarks vs. distance to manual landmarks

landmark is considered as correctly detected if its distance to manual landmark point is less than 8 mm. The nose tips and points under the nose are well estimated with 98.4% detection rate. The landmarks on mouth corners and middle of chin are relatively poorly detected, with results of less than 90%. The reason for poor detection rate of these landmarks may be the expression variations in the testing set. This landmark estimation error may result in a performance decrease in face recognition.

Table 4.3 compares the performance of the proposed face recognition algorithm on the 3D face images that are registered: using manually labeled landmark points, automatically via ICP and using automatically detected landmark points. The depth images were generated by ray-casting and DCT-4 feature was used for classification. The results obtained using the automatic registration methods -with ICP and with automatically detected landmark points-, are slightly lower than the results obtained on the images that are registered using the manual labels. The decrease in the performance is mainly caused by the errors introduced by ICP registration and automatic landmark detection. Better results were attained on the images that are registered using automatically detected landmark points than on the ones registered via ICP. This indicates that deformation onto a common frame is able to mitigate the effects of expression variations.

Landmark	Detection Rate
Out left eye corner	93.1%
Inner left eye corner	97.8%
Saddle point	97.3%
Inner right eye corner	98.1%
Out right eye corner	91.3%
Tip of nose	98.4%
Point under nose	98.4%
Left mouth corner	85.1%
Middle of mouth	89.7%
Right mouth corner	83.9%
Middle of chin	82.6%

Table 4.2: Detection rate of landmark points.

Registration methods	Recognition Rate
Manual LM_10	95.5%
ICP	92.7%
Automatic LM_10	93.1%

Table 4.3: Manual registration vs. automatic registration.

Methods	Manual LM_10	Automatic LM_10
LocalDCT	95.5%	93.1%
Local DCT+SVM	90.0%	89.0%
EHMM	87.9%	85.5%
Eigenfaces	88.6%	86.5%
LDA	92.4%	88.5%
Bayesian	94.9%	89.7%
PSD	81.4%	80.6%
PDM	87.6%	84.7%

Table 4.4: Perfomance comparison of methods with manual and automatic landmark based registration.

4.4 Performance Comparison

In this part of the experiments, we compare the proposed local appearance-based 3D face recognition approach with several well-known face recognition algorithms: Eigenfaces [33], linear discriminant analysis (LDA) [37], Bayesian face recognition [27], embedded hidden Markov model (EHMM) [28], point set difference (PSD) [24] and point distribution model (PDM) [10]. We also used SVM classifier instead of nearest neighbor classifier to assess the performance of a more sophisticated classification scheme in a local appearance face recognition framework.

Table 4.4 shows the experimental results of each algorithm. Both of the correct recognition rates attained on manually and automatically registered images are given. In Eigenfaces, Bayesian face recognition and PDM algorithms we used 100 principal components. This is the number of principal components that we achieved the best results with. For LDA we used the LDA+PCA algorithm provided in the CSU face identification evaluation system [11]. This version of LDA uses a soft distance measure proposed by Zhao et al. [37]. In EHMM, we used a 4×4 size DCT coefficients matrix as an HMM observation, which is extracted from a 12×12 image block by taking DCT. In local appearance 3D face recognition, both for nearest neighbor and SVM classification, we used ten dimensional DCT-4 feature set. Radial basis function was the kernel function in SVM classifier.

From the results given in Table 4.4 it can be observed that the proposed local appearance approach outperforms the other well-known face recognition algorithms as well as the local DCT features classified with SVM, which may suffer from small training sample problem. All of the algortihms' performance decreases slightly when they use the images that are registered using automatically detected landmarks. These results indicate that the proposed local DCT features provide a powerful and robust representation of the depth images for the classification purposes.

Chapter 5

Conclusions

In this paper, we proposed a depth image based 3D face recognition approach using local appearance-based models. Depth images were obtained by a base mesh-based registration and a resampling technique. We extracted the local features from each block on a depth image using discrete cosine transform, and then concatenated the local features in order to conserve spatial information.

We conducted extensive experiments on the range images from the FRGC version 2 face database to investigate several factors that may affect the recognition performance. First, we performed face recognition experiments using different local feature dimensionality. We observed that the correct recognition rate increases at the beginning with growing feature vector dimensionality. It reaches the best result with ten-dimensional local feature. After this point, the recognition rate remains the same or decreases slightly. The second experiment investigated different feature sets and frequency contents. We found that DCT-4 feature set, which excludes the average horizontal and vertical depth changes as well as the average depth value, attains the best results. Third, we analyzed landmark selection for TPS-based registration to improve dense correspondence. In the experiments, we obtained the best result by discarding the landmark located on the middle of the mouth. During these experiments, we also observed that, ray-casting mapping always outperforms closest-point mapping when face surfaces are resampled with the base mesh. In the last experiment, we assessed the performance of the fully automatic systems. We compared two automatic registration methods, rigid ICP and non-linear TPS warping based on automatic landmarking. Face recognition performance decreased slightly with automatic registration. However, having only a slight reduction in the correct recognition rate due to automatic registration indicated that, it is possible to have an online fully automatic 3D face recognition system without sacrificing much performance. We also thoroughly compared the proposed local appearance-based algorithm with the well-known face recognition algorithms (PCA [33], LDA [37], PDM [10], PSD [24], Bayesian [27], EHMM [28]) as well as with the local appearance-based approach using SVM classifier. Experimental results showed that the proposed algorithm provides an improvement over existing algorithms in face recognition performance.

Bibliography

[1] L. Akarun, B. Gökberk, and A.A. Salah. 3D Face Recognition for Biometric Applications. *13th European Signal Processing Conference (EUSIPCO)*, 2005.

[2] D. Akca. Generalized Procrustes Analysis and its Applications in Photogrammetry. Technical report, IGP-ETH, Zurich, May 2003.

[3] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *European Conf. Computer Vision*, 1996, pp. 45-58.

[4] P.J. Besl, R.C. Jain. Three Dimensional Object Recognition. *ACM Computing Surveys*, Vol. 17, pp. 75-145, 1985.

[5] F.L. Bookstein, Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *Psychometrika*, Vol. 11, No. 6, pp. 567-585, June 1989.

[6] K.W. Bowyer, K. Chang, and P. Flynn. A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition. *Computer vision and image understanding*, Vol. 101, pp. 1-15, 2006.

[7] A. Bronstein, M. Bronstein, and R. Kimmel. Expression Invariant 3D Face Recognition. Proc. of *Audio and Video-based Biometric Person Authentication*, 2688.

[8] K. Chang, K.W. Bowyer, and P. Flynn. Multi-Modal 2D and 3D Biometrics for Face Recognition. Proceedings of *the IEEE International Workshop on Analysis and Modelling of Faces and Gestures*, pp. 187-194, 2003.

[9] C.S. Chua, F. Han, and Y.K. Ho. 3D Human Face Recognition Using Point Signatures. Proceeding of Int. Conf. on *Automatic Face and Gesture Recognition*, pp. 233-237, 2000.

[10] T.F. Cootes, C.J. Taylor, and D.H. Cooper, and J. Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, Vol. 61, No. 1, pp. 38-59, 1995.

[11] The CSU Face Identification Evaluation System: http://www.cs.colostate.edu/evalfacerec/.

[12] H.K. Ekenel, Q. Jin. ISL Person Identification System in the CLEAR Evaluations. *CLEAR Evaluation Workshop*, 2006.

[13] H.K. Ekenel, A. Pnevmatikakis. Video-Based Face Recognition Evaluation in the CHIL Project - Run 1. *7th International Conference Automatic Face and Gesture Recognition (FG2006)*, 2006.

[14] H.K. Ekenel, R. Stiefelhagen. A Generic Face Representation Approach for Local Appearance based Face Verification. *CVPR Workshop on FRGC Experiments*, 2005.

[15] H.K. Ekenel, R. Stiefelhagen. Analysis of Local Appearance-based Face Recognition: Effects of Feature Selection and Feature Normalization. *CVPR Biometrics Workshop*, NYC, USA, 2006.

[16] H.K. Ekenel, R. Stiefelhagen. Local Appearance based Face Recognition Using Discrete Cosine Transform. *13th European Signal Processing Con ference (EUSIPCO 2005)*, Antalya, Turkey, 2005.

[17] R.C. Gonzales, R.E. Woods. Digital Image Processing. Prentice Hall, 2001.

[18] G. Gordon. Face recognition based on depth and curvature features. In Proc. *IEEE CVPR*, pp. 108-110, 1992.

[19] R. Gottumukkal, V.K. Asari. An improved face recognition technique based on modular PCA approach. *Pattern Recognition Letters*, 25(4), March 2004.

[20] J. Gower. Generalized Procrustes Analysis. *Psychometrika*, Vol.40, No. 1, pp. 33-51, 1975.

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

[22] C. Hesher, A. Srivastava, and G. Erlebacher. A novel technique for face recognition using range imaging. in Proc. *7th Int. Symposium on Signal Processing and Its Applications*, pp. 201-204, 2003.

[23] T. Hutton, B. Buxton, and P. Hammond. Dense Surface Point Distribution Models of the Human Face. Proceedings of IEEE Workshop on *Mathematical Methods in Biomedical Image Analysis*, pp. 153-160, Hawaii, 2001.

[24] M.O. İrfanoğlu, B. Gökberk, and L. Akarun. 3D Shape-Based Face Recognition Using Automatically Registered Facial Surfaces. in Proc. *ICPR*, vol.4, pp. 183-186, 2004.

[25] X. Lu, D. Colbry, and A.K. Jain. Three Dimensional Model Based Face Recognition. in Proc. *ICPR*, 2004.

[26] X. Lu, A.K. Jain. Deformation Analysis for 3D Face Matching. In Proc. *IEEE WACV*, 2005.

[27] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian Face Recognition. *Pattern Recognition*, Vol. 33, No. 11, pp. 1771-1782, November, 2000.

[28] A. Nefian. A Hidden Markov Model-based Approach for Face Detection and Recognition. PhD thesis, Georgia Institute of Technology, 1999.

[29] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *CVPR'94*, 1994.

[30] A. Srivastava, X. Liu, and C. Hesher. Face Recognition Using Optimal Linear Components of Range Images. *Image and Vision Computing*, Vol. 24, No. 3, pp. 291-299, 2006.

[31] H.T. Tanaka, M. Ikeda, and H. Chiaki. Curvature-Based Face Surface Recognition Using Spherical Correlation - Principal Direction for Curved Object Recognition. Proceeding of Int. Conf. on *Automatic Face and Gesture Recognition*, pp. 372-377, 1998.

[32] F. Tsalakanidou, D. Tzovaras, and M. Strintzis. Use of Depth and Colour Eigenfaces for Face Recognition. *Pattern Recognition Letters*, Vol. 24, pp. 1427-1435, 2003.

[33] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Science*, pp. 71-86, 1991.

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

[35] Y. Wang, C. S. Chua, Y.K. Ho, Facial Feature Detection and Face Recognition From 2D and 3D Images, *Pattern Recognition Letters*, Vol. 23, pp. 1191-1202, 2002.

[36] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. Face recognition: A literature survey. ACM computing surveys 35:44, 399-458, *Association for Computing Machinery*, 2003.

[37] W. Zhao, R. Chellappa, and P.J. Phillips. Subspace linear discriminant analysis for face recognition. In *UMD*, 1999.