ABSTRACT

MIAO, SHUN. 3D Face Recognition From Range Image. (Under the direction of Professor Hamid Krim).

In this thesis, we explore the statistical and geometrical behavior of the uncontrolled parameters of a human face, including both the rigid transform caused by a head pose and the non-rigid transform caused by a facial expression. We focus on developing a 3D facial recognition scheme which is robust for these uncontrolled parameters.

This thesis presents a novel 3D face recognition method by means of the evolution of iso-geodesic distance curves. Specifically, the proposed method compares two neighboring iso-geodesic distance curves, and formalizes the evolution between them as a one-dimensional function, named evolution angle function, which is Euclidean invariant. The novelty of this paper consists in formalizing a 3D face by an evolution angle functions, and in computing the distance between two faces by that of two functions. Experiments on Face Recognition Grand Challenge (FRGC) ver2.0 shows that our approach works very well on the neutral faces. By introducing a weight function, we also show a promising result on a non-neutral face database.

A novel 3D surface segmentation scheme is developed to detect the partial similarity between two 3D facial images. The proposed algorithm is based on the iterative closest point (ICP) algorithm, which uses the mean square distance as the cost function and is not able to detect partial similarities. The presented thesis make an improvement of the ICP algorithm by iteratively removing points contributing the largest error, and the remaining area of surface can be shown to be the partial similarity between two surfaces.

3D Face Recognition From Range Image

by
Shun Miao

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fullfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh, North Carolina

2010

APPROVED BY:

Dr. Griff Bilbro

Dr. Wesley Snyder

Dr. Hamid Krim
Chair of Advisory Committee

DEDICATION

For my mother *Yaling Wen* and my father *Qinan Miao*, who have always been there giving me love and support.

BIOGRAPHY

Shun Miao was born in Jan. 15th 1987, to Qinan Miao and Yaling Wen, in Qingdao, China. He received his bachelor's degree in Electrical Engineering in 2009, from Zhejiang University, China. By a collaborate program between Zhejiang University and NC State University, he came to ECE department of NC State University for his graduate study in August, 2008. After one semester in NC State, he fortunately joined Professor Hamid Krim's Vision, Information and Statistical Signal Theories and Applications group (VISSTA), and started working on 3D face recognition research.

Shun's research focuses on pattern recognition, especially the application on face recognition. He currently emphasizes on 3D surface segmentation for detecting partial similarities.

Shun is member of the Institute of Electrical and Electronics Engineers (IEEE) since 2009.

ACKNOWLEDGMENTS

This master degree involves more than one's brain. I wish to leave some words to reflect on the human side of my two years journey; and acknowledge all the people who contributed to the completion of this work.

My greatest thanks to my parents, who were always staying behind me, giving me as much support as they could. Father, I would never forget that you flied hundreds of miles, merely for the purpose of sitting beside a professor to inquire about educating children. And mother, it is extraordinary all those full nights you spent beside my hospital bed, giving me brave to recover from the injure. Your support helped to me go all the way to US, and to complete my Master's work. Thank you for your love and support.

I would like to thank my advisor, Professor Hamid Krim, for all your guidance in my study and research, and your support on my PhD application. You are always very patient to listen to your students and give us valuable suggestions. I believe that what I have learned from working with you will be helpful in my whole life. Besides, you've given us extraordinary valuable opportunities to meet, on semimonthly research seminar, with world renowned scientist. Those seminar lectures helped me a lot on conducting my research. I would also like to acknowledge Professor Wesley Snyder, who took his valuable time to meet with me weekly, giving me very helpful suggestions. And thank you Professor Griff Bilbro, for you joining my advisory committee. I sincerely thank to professors who have been part of my Master program, alphabetical, Prof. Winser Alexander, Prof. Brian Hughes, Prof. Xiaobiao Lin , Prof. Larry Norris and Prof. J. Keith Townsend.

I also thank to all my colleagues in VISSTA lab. I have a fantastic time working with you guys, with a special thanks to Sheng Yi. You are intelligent, friendly and very helpful. I would also like to thank all nice people I've met during my journey in US, but I afraid the number of them is to high to be possibly exhaustively listed. I will remember every smiling face of you guys, and thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Face recognition is a challenging task that has been extensively researched during the last decade[1]. While most of previous face recognition works were based on 2D images, as the development of 3D scanning techniques, 3D face recognition has gained more and more attention. In a controlled environment, current 2D face recognition techniques can achieve an acceptable accuracy, but inadequate for more challenging applications, especially in the presence of variation of lighting conditions and poses [1]. 3D face recognition, by including the complete geometrical information, provides a potential to alleviate the impact of lightening and pose, and to therefore improve recognition performance.

Although more promising, the use of 3D images for face recognition also presents some challenges. First, as a surface in cartesian coordinates, a face is always subjected to Euclidean transformations, and a face alignment stage is usually required before comparison, entailing additional computational cost. Second, in most scenarios, the 3D images we have are represented by random samples on the face, or triangulation without consistent parameterizations. As a result, 3D face comparison becomes a challenging task.

In this thesis, we explore the statistical and geometrical features of human faces under different facial expressions. Our objective is to come up with a face recognition scheme based on range images to achieve a good recognition rate under the variety

of facial expressions. We proceeded to extract low dimensional features from facial range image, and built up a classifier, which can handle the deviations caused by uncontrolled parameters, such as a pose of head and a facial expression.

1.1 Motivation and Overview

In the first work of the presented thesis, we focus on modeling the evolution of facial level set curves named iso-geodesic curves, whose level set function is defined as the geodesic distance to a reference point. Our interest in the evolution of iso-geodesic curves is motivated on two counts. First, it has been shown that, change of a geodesic distance on a human face caused by facial expressions is much less than change of a Euclidean distance, and can hence be treated as an isometric transform. Therefore, the registration based on the geodesic distance is robust. Secondly, while the 3D curves are represented by cartesian coordinates that depend both on a reference and on a base, the evolution of these curves only contains intrinsic information and is Euclidean invariant. In the end, discriminant analysis is applied to evaluate the robustness of these extracted features under different facial expressions and the robust ones are selected for recognition.

The second part of the work in this thesis is focusing on 3D surface segmentation. We attempt to extract identical areas from two 3D human faces, and exploit them to perform recognition. We believe, with experimented validation, that two facial images of the same subject, share a considerably large portion of the facial area, even if these two images have different facial expressions. And for two images with different subjects, it is very unlikely to obtain two large identical areas from them. Partial similarity detection is however a very challenging task because 3D images are subjected to Euclidean transformations, and correct rotation and translation parameters are required to overlap the identical areas with little prior knowledge. In the face segmentation scenario, we have the prior knowledge that a large facial area will remain invariant to most facial expressions of interest. The iterative closest point may be applied to obtain the rotation and translation, which minimize the global mean

square distance. We shrink the surface iteratively by removing nodes contributing large errors, until the remaining patches of the surfaces are identical. The area of matched surfaces is used as a similarity measure of these two facial images. The more area can be fitted, the more likely these images are from the same subject.

1.2 Contribution

We summarize our main contributions below.

- We have investigated the representation, classification and recognition of 3D faces under variance of facial expressions. We propose to analyze the evolution of iso-geodesic distance curves to obtain features that contain only the intrinsic geometrical information of human faces. Features of 3D faces used in the presented thesis also have a nice property that they are local Euclidean invariants of a 3D surface. With the surface registration based on iso-curves, we are finally able to apply discriminant analysis techniques to select features that are robust for facial expressions.

- We introduce a 3D surface segmentation technique for searching partial similarities between two facial images. With the empirical knowledge that human faces keep significant partial similarities under facial expressions, the segmentation scheme is applied to locate the invariant area between the same person's two facial images with different facial expressions. The key idea in this problem is to iteratively expand or shrink the surface under the constrain of a mean square distance.

1.3 Outline

The present thesis is organized as follows.

- Chapter 2 gives a literature review of the state of art in face recognition. The main 3D facial recognition techniques, with their experimental results are pre-

sented. One may distinguish these main approaches as curve cased, non-rigid surface based and template matching based approaches.

- Chapter 3 introduces a 3D face recognition algorithm based on evolution of isogeodesic distance curves. Both theoretical and numerical results are described in this chapter.

- Chapter 4 demonstrates a 3D surface segmentation based on the Iterative Closest Point (ICP) algorithm. An improvement is made on the ICP algorithm to enable it to register partial similar surfaces.

- Chapter 5 proposes a novel framework of 3D surface segmentation as a future work. The proposed framework combines ICP algorithm and level set method, for the purpose of detecting facial regions which is invariant to the presented facial expression.

Chapter 2

Background

Research interest in 3D face recognition has witnessed an explosive growth over the last few years. At first, people attempted to use 3D data to rule out variations of illumination and pose, and therefore enhance face recognition approaches based on 2D images [2][3][4][5]. There are also several works on directly applying existing 2D recognition techniques on depth images. In this chapter, we focus on the most recent and important face recognition techniques. They are categorized into three themes, a curve based, a non-rigid surface based and a template matching based approaches.

Because a 3D surface is subjected to a Euclidean transformation, most 3D face recognition approaches require surface matching. A well known surface matching method is the Iterative Cloud Point (ICP) algorithm [6][7]. Feature points on faces are also widely used in surface alignment [8][9][10][11][12]. The difficulty of surface matching is clear. As human faces are not rigid surfaces, deformation will severely undermine the performance of most algorithms.

Some new ideas have recently been provided for non-rigid matching. Bronstein *et al.* proposed to match non-rigid surface in an isometric manner. Lu *et al.* [13] performed a Thin-Plate Spline warping to establish a deformable face model.

To avoid a consuming and difficult surface matching procedure, researchers have explored approaches based on analysis of facial curves. Samir *et at.* [14] used level set curves of a height function to study the difference between surfaces. Shuo *et al.* [15]

compared level set curves of geodesic distance function by developing a Euclidean invariant. Srivastava *et al.* [16] measured the distance between two facial curves on a shape manifold.

Another way to avoid surface matching is to fit a template on face data. A 3D face morphable model [17] was proposed by Blanz *et al.*, an active appearance model was proposed by Cootes *et al.* [18] and an annotated deformable model was proposed by Passalis *et al.* [19] are examples.

2.1 Curve Based Approaches

Samir *et al.* [14] presented a 3D face recognition method, which measures the similarity of faces on the manifold of closed planar curves. Level set curves of a depth function are extracted from 3D faces, and parameterized by arc length. A manifold of closed, arc length parameterized planar curves is generated, on which an approach of computing a geodesic on the manifold is proposed. The geodesic length between two curves on the shape manifold is used as a criterion in the nearest neighborhood recognition as the measurement of similarity. In the validation experiment, using a data set of 300 facial surfaces, 6 facial expressions each for 50 subjects, a recognition rate of 92% is reported. The method is based on depth level set curves, but depth is subjected to Euclidean transform, thus a costly alignment procedure is required.

Shuo *et al.* [15] proposed a Euclidean invariant for curves in R^3 to compare geodesic circles from 3D faces. Level set curves of a geodesic distance between any point on a face surface to nose tip are extracted, and are referred to as geodesic circles. A Euclidean integral invariant signature for curves in R^3, which is independent for positions and parameterizations of curves, is presented. Based on the assumption that geodesic circles undergo piecewise Euclidean transformations on account of facial expressions, the approach is claimed to be robust to facial expressions. The L^2 norm of integral signature is computed, and used as the distance in a Nearest Neighbor classification. In the experimental evaluation of this work, training data are selected from FGRC Spring 2003, and testing data from FGRC spring 2004. A recognition

rate of 95% is reported.

Mpiperis *et al.* [3] proposed to use geodesic polar coordinates to register faces and map the texture to the same polar system. The geodesic polar coordinates (r, θ) are defined as: radius r is the geodesic distance to nose tip, angle θ represents the tangent direction of a geodesic path at nose tip. A mapping from 3D faces to a planar surface is defined based on the geodesic polar coordinates (r, θ). The mapping $f : S_1(r, \theta) \rightarrow S_2(r, \theta)$ is claimed to be an isometry, but which, unfortunately, is not generally true. One experiment embeds the texture on a 3D face to a geodesic plane, and uses the flattened image for classification. Another experiment embed the depth to geodesic plane and perform classification. Experimental results are from 80.3% to 90.4% rank-one recognition rate using texture, and from 84.4% to 95% using depth, depending on the choice of database.

Srivastava *et al.* [16] used the geodesic distance between geodesic circles on a shape manifold for curves in arbitrary R^n to measure the similarity between 3D faces. The pre-shape space, as a quotient space with respect to re-parametrization and rotation, is constructed using square-root velocity function [20][21], which can represent a curve in R^n by n functionals. An algorithm to compute a geodesic between two parameterized curves is introduced, and the geodesic distance serves as the similarity measurement for face recognition. Tested on 12 faces of 7 subjects, a recognition rate of 100% is achieved.

2.2 Non-rigid Surface Based Approaches

Bronstein *et al.* [22][23] proposed an isometric model of facial expressions to allow for deformation caused by facial expressions. Based on the assumption that "the change of the geodesic distance due to facial expressions is insignificant", an isometric embedding is proposed to embed 3D face in R^3, keeping the Euclidean distance in the embedded space equal to the geodesic distance on the original surface. Face data is converted by this embedding to a canonical form, and the L^2 norm of the canonical norm is used for classification. Tested on a data set with 220 images of 30 subjects,

a recognition rate of 100% is reported.

Li *et al.* proposed to use multiple features for 3D face recognition. In their first attempt [24], multiple intrinsic geometric features, including angles, geodesic distance and curvatures are extracted, and a training procedure is proposed to obtain weights for extracted features, according to their sensitivity to facial expressions. And a nearest neighbor classifier is applied using the training results as a basis. Experimental evaluation is carried out using a dataset containing 300 frontal faces, 5 faces per subject, of all the 60 subjects. The training group contains 150 faces of 30 individuals, and the test group contains the remaining 150 faces. The result showed a recognition rate of 94.17%, while rigid surface approach reaches a recognition rate of 87% on the same dataset. In their second attempt [25], a uniform remeshing scheme is proposed to extract multiple features, and a ranking procedure is applied to select expression-insensitive features. The test on the same dataset reaches a recognition rate of 94.68%.

Lu *et al.* [13] improve previous work using a Thin-Plate Spline (TPS) approach for deformation analysis [26] to a deformable model which is able to synthesize different facial expressions from a given neutral face. 94 fiducial landmarks on 3D faces are extracted from both neutral faces and non-neutral faces in the training set, three of which are used for alignment. Given a probe face, the displacements of landmarks are added to the face, and TPS warping is applied to interpolate the deformation of landmarks to the whole surface. Tested on 877 face scans and 100 gallery templates, trained by 7 independent facial expressions, a performance of 92% rank one recognition rate is achieved.

2.3 Template Matching Approaches

Cootes *et al.* introduced a active appearance model. Shapes of 3D faces are normalized by a Thin-Plate Spline warping acted on extracted facial anchor points. Principal Component Analysis is applied both on shape and texture to extract coefficients, and an iterative model refinement algorithm is developed to fit the model

to face scans. The quality of the model is evaluated by the number of iterations and surface displacement. Edwards *et al.* [27] use this face model for face recognition. In a test with 400 faces, 200 for training and 200 for testing, a recognition rate of 88% is reported.

Similar to the idea of an active appearance model, Blanz *et al.* [17][28][29][30] proposed to use a 3D morphable model [31] for face recognition in both 2D and 3D. The correspondence between 3D faces and a reference face is established by the optical flow, prior to applying PCA on the reference shape and texture vectors to generate the morphable model. The illumination model of Phong is adopted to synthesize different illuminations and colors. This morphable model can reconstruct a 3D model from single 2D image. The experimental evaluation in this work uses a database with front, side and profile images in both gallery set and probe set, for a recognition rate of 99.8% for frontal-frontal case, and the lowest recognition rate of 79.5% occurs in the profile-frontal case. Huang *et al.* [32] added a component based approach to the morphable model, but the recognition rate remains the same.

Xu *et al.* [33] developed an automatic subdivision technique for consistency of a mesh. They fit a square mesh on 3D face and repeat subdivision to refine the base mesh. Gaussian-Hermite moment of shape vector is computed as feature for Nearest Neighbor classifier. Tested in a manually fitted database with 30 subjects, 6 images for each, their approach achieves a recognition rate of 96.1%. However, tested with a automatically fitted database with 60 subjects, 6 images for each, the recognition rate decreases to 72.4%.

Passalis *et al.* [19][34] presented an approach to 3D face recognition that use Haar wavelet coefficients on annotated deformable models (AFM). AFM is constructed using the average 3D facial meshes from a training set, and UV parametrization of AFM is used to generate a three-channel deformation image, where each channel value records one of three position coordinates. A four level Walsh transform using a Haar wavelet is performed on each channel, to extract coefficients that will be used for comparison. The experimental evaluation in this work uses 466 images in the gallery set and 3541 images in the probe set, while 60% of subjects have a neutral

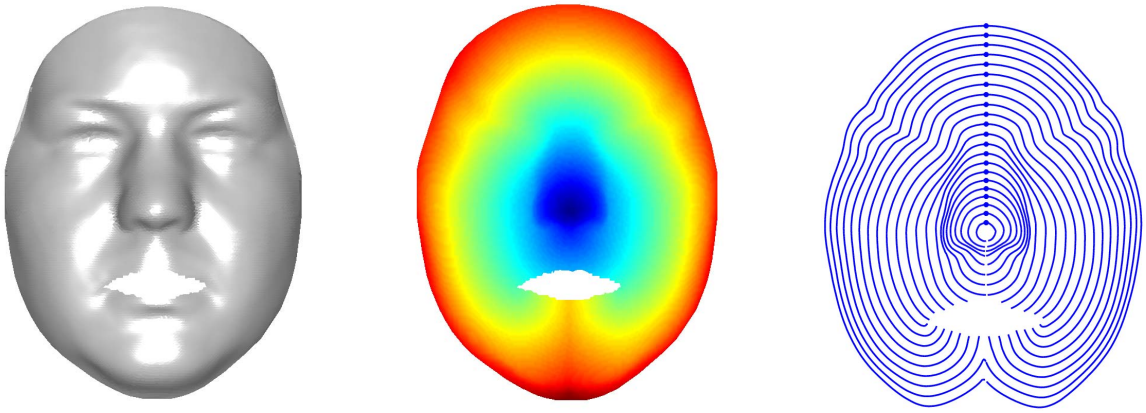
facial expression and 40% of subjects have a variety of non-neutral facial expressions. With a false alarm rate of 10^{-3}, a verification rate of 95.43% is reported on neutral subsets, and 80.61% on non-neutral subset.

Chapter 3

Face Recognition Using Facial Curves

In this chapter, we develop a novel face recognition technique based on range images. We model the evolution of level set curves on a 3D surface to derive 3D Euclidean invariant features. Discriminant analysis is applied to these features to extract facial expression-insensitive features, which are used in a nearest neighbor classifier for recognition.

The remainder of this chapter is organized as follows. In Section 3.1, we introduce the background of iso-geodesic distance curves, as well as an efficient algorithm to extract it. In Section 3.2, we propose a 3D Euclidean invariant evolution angle function, which is shown to be Euclidean invariant and to contain the full information of a 3D surface. In Section 3.3, details about the implementation of the algorithm are provided. In Section 3.4, expression-insensitive features are extracted by performing a discriminant analysis. Experimental results and conclusions are presented in Section 3.5.



(a) (b) (c)

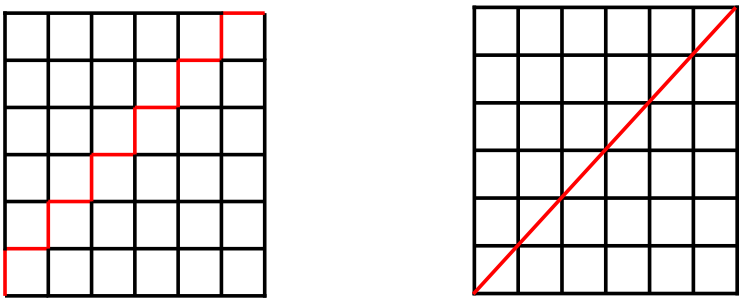
Figure 3.1: (a) Smoothed 3D face (b) Geodesic Distance Function (c) Iso-curves, starting point is marked on each curve. To be clear

3.1 Geodesic Distance Function and Iso-curves

The geodesic distance between two points on a manifold is defined as the arc length of the shortest path between these two points on the manifold. Given a 3D face M and the nose tip $p_{nose} \in M$, the Geodesic Distance Function (GDF) $G_M(p)$ is a mapping from M to R defined as $G_M(p) = D_M(p, p_{nose})$ where $D_M(\cdot, \cdot)$ denotes the geodesic distance between two points on the manifold M. If we connect points with the same GDF value, we will obtain a set of curves, denoted as iso-geodesic distance curves (iso-curve), which are also level set curves of GDF. The iso-curve with GDF value t can be written as (see Fig. 3.1 (a) and (b) for example)

$$C(t) = \{p \,|g_M(p) = t, p \in M\}\,. \tag{3.1}$$

To extract iso-curves, we need to compute GDF first. There are several numerical method for computing distances on a triangular mesh. Since our data is stored as 3D point clouds, which can easily be converted to a graph, the Dijkstra algorithm can be applied to compute on-graph distance between nodes. However, the distance obtained by Dijsktra algorithm is usually not precise. The red path in Fig. 3.2 (a) is the geodesic path obtained from Dijsktra algorithm, which is obviously subjected



(a) (b)

Figure 3.2: (a) Geodesic path obtained from Dijstra algorithm (b) Geodesic path obtained from Fast Marching

to error because the path is strictly constrained on the graph. One solution is to use the Fast Marching Algorithm [35] to compute the on-surface geodesic path and distance as shown in Fig. 3.2 (b). The geodesic distance computed by fast marching algorithm is the real shortest path on a triangulated surface, and is consistent to the real geodesic distance.

The GDF computed by fast marching algorithm is shown in Fig. 3.1 (b). Because the GDF we have is discrete, and only assumes values on vertices of a triangular mesh, we obtain an iso-curve with level C, by an approximation and interpolation. The strategy is to first obtain an edge across the t-level iso-curve, and then interpolate the edge to obtain a point. If an edge across the t-level iso-curve, it is obvious that the GDF value at one node of the edge is smaller than t, and the GDF value for the other node is greater than t. Therefore, for an edge starting from p_1 and ending at p_2, the product

$$S(p_1, p_2) = (G_M(p_1) - t)(G_M(p_2) - t) \tag{3.2}$$

indicates whether this edge is across the t-level iso-curve. If $S(p_1, p_2)$ is negative, which means two nodes of the edge locate at two side of the t-level iso-curve, then the edge intersects with the iso-curve, otherwise, it does not. For all edges intersecting

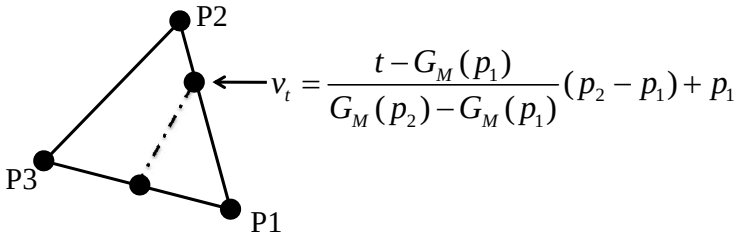


Figure 3.3: Linear interpolation for detecting iso-curves

iso-curves, the intersection is linearly interpolated as [36] (see Fig 3.3 for illustration).

$$v_t = \frac{t - G_M(p_1)}{G_M(p_2) - G_M(p_1)}(p_2 - p_1) + p_1. \tag{3.3}$$

3.2 Euclidean Invariant Evolution Angle Function

Essentially, iso-curves are level set curves defined on a manifold, with GDF as a level set function. In this section, we will discuss the evolution of these iso-curves from the perspective of differential geometry as the level value t increases. In Section 3.2.1, the direction of the curve evolution is analyzed and evolution vector function (EVF) is proposed to represent the curve evolution. In Section 3.2.2, by utilizing a nice property that iso-curves evolve at a constant speed, evolution angle function (EAF) is proposed to represent the 3D face in a Euclidean invariant way.

3.2.1 Evolution Vector Function

Iso-curves, as space curves, can be written as

$$C(s,t) = \big(x(s,t), y(s,t), z(s,t)\big) \tag{3.4}$$

where t denotes the level value of the geodesic distance to nose tip, and s the parameter of the curve. Let's consider the tangent plane at the point $C(s,t)$, which

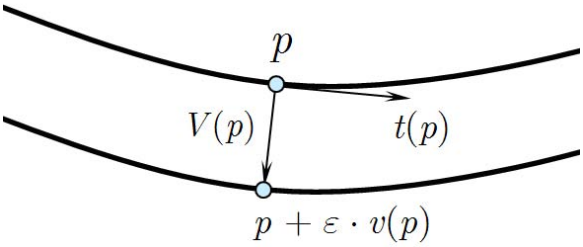


Figure 3.4: Illustration of evolution of iso-curves.

is spanned by two orthogonal vectors. The first vector is the tangent vector of the iso-curve at the point $C(s,t)$, denoted by $T(s,t)$

$$T(s,t) = \Big(\frac{d}{ds}x(s,t), \frac{d}{ds}y(s,t), \frac{d}{ds}z(s,t)\Big) \tag{3.5}$$

The other vector is orthogonal to $T(s,t)$, denoted by $B(s,t)$. $T(s,t)$, $B(s,t)$ and $N(s,t)$ formalize a basis of three-space, where $N(s,t)$ denotes the normal to the surface.

The evolution speed vector $E(s,t)$ can be decomposed into these three directions as $E(s,t) = \alpha T(s,t) + \beta B(s,t) + \gamma N(s,t)$. Because iso-curves are constrained to the surface, the normal component γ is obviously equal to zero. The tangential component can also be set to zeros since tangential movement does not cause any change of t and it only affects the parameterization of the curve. Therefore, the evolution speed vector can simply be set to the binormal component $\beta B(s,t)$, and according to the definition of an iso-curve the speed is $\beta = 1$. For simplicity, from now on, the evolution speed vector is denoted by $v(s,t)$, and $|v(s,t)| = 1$.

If we extract iso-curves $c_i(s) = C(s, t = i\Delta)$ with a small enough Δ, the movement along direction $v_i(s)$ causes the same amount change in GDF (see Fig.3.4 for example):

$$f(p + \Delta \cdot v(p)) = f(p) + \Delta \tag{3.6}$$

The speed vector $v_i(s)$ is a function of arc length s. By multiplying the time interval Δ, we can obtain an evolution vector function (EAF) $v_i(s)\Delta$, which is an

approximation of evolution between the i-th iso-curve and the $(i+1)$-th one. This is an *approximation* because the speed vector varies in the interval Δ, and $v_i(s)\Delta$ is the approximated evolution vector based on the assumption that $v_i(s)$ is a constant in the interval Δ. However, when Δ is small enough, the approximation is quite precise and the evolution can be written as Eq.3.7. The $(i+1)$-th iso-curve has a different parameter $\hat{s}$ because evolution cannot guarantee a preservation arc length parameterization.

$$c_i(\hat{s}) = c_i(s) + v_i(s) \cdot \Delta \tag{3.7}$$

3.2.2 Evolution Angle Function

Since evolution vectors are subjected to Euclidean transformations, they can hardly be used for recognition. Using the fact that the magnitude of an evolution vectors is always equal to one, we are able to represent the evolution in a Euclidean invariant way. Because in the (T, B, N) frame, the tangential component is always zero, which means the evolution vector is in a plane that is perpendicular to the tangent vector. Because $v_i(s) \equiv 1$, the $v_i(s)\Delta$ lives on a circle with radius Δ and perpendicular to the tangent vector. If we are given a reference vector on this unit circle, $v_i(s)$ can be determined by an angle, which is Euclidean invariant. To generate a set of reference vectors along an iso-curve, we establish a moving frame $\{R_i(s), S_i(s), T_i(s)\}$ along it, where $T_i(s)$ is the tangent vector which can be computed by the curve itself. Vectors from the nose tip to the iso-curve are denoted as $q_i(s)$, and the other two vectors are defined as:

$$\begin{aligned} R_i(s) &= \frac{q_i(s) - q_i(s)\frac{T_i(s)}{||T_i(s)||}}{||q_i(s) - q_i(s)\frac{T_i(s)}{||T_i(s)||}||}, \\ S_i(s) &= \frac{R_i(l) \times T_i(l)}{||R_i(l) \times T_i(l)||} \end{aligned}$$

Then, the unit evolution function can be written as a function of angle $\theta^i(l)$, which

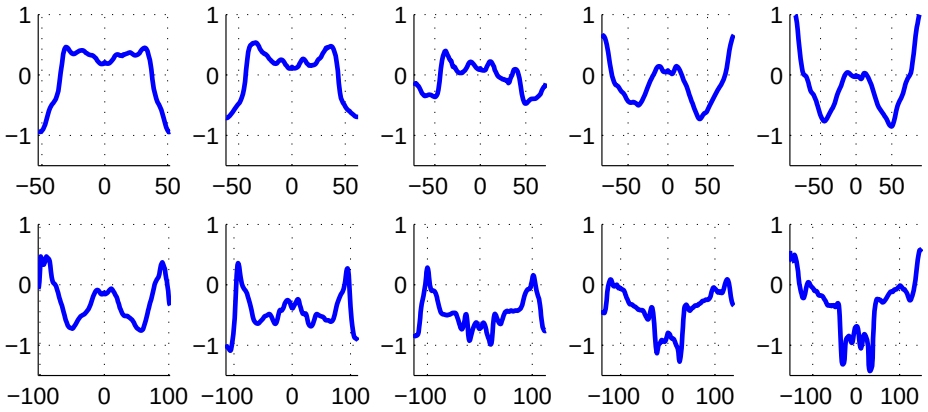


Figure 3.5: Examples of Evolution Angle Function at different level

is called an evolution angle function (EAF):

$$v_i(s) = R_i(s)\cos(\theta_i(s)) + S_i(s)\sin(\theta_i(s)) \tag{3.8}$$

By substituting Eq. 3.8 to Eq. 3.7, the evolution between two neighboring iso-curves is explicitly determined by EAF, which means EAFs include the necessary information needed to reconstruct the iso-curves. Given a initial Now, the problem of comparing two faces has been reduced to comparing one-dimensional functions, which is much easier because we don't need to consider surface alignment and registration any more. A set of evolution angle functions is shown in Fig. 3.5.

3.3 Implementation

3.3.1 Face Detection and Preprocessing

3D images used in this thesis are collected from FRGC ver2.0 database [37], with 640×480 face range images and 2D images. Raw images are first down sampled by 4 in both X and Y direction to reduce the resolution. An image is represented by 6 160×120 matrix $\boldsymbol{(X, Y, Z, R, G, B)}$, where $\boldsymbol{X, Y, Z}$ denote the x, y, z coordinates

of each vertex and $\boldsymbol{R, G, B}$ denote the coordinate in color space. After locating the nose tip, all vertices with distance to nose tip less than 100mm are selected to formalize a point cloud as our region of interest. Delaunay triangulation is applied to $\boldsymbol{(X, Y)}$ to generate a triangular mesh.

3.3.2 Curve Parameterization

It is obvious that Equation 3.3 only provides one point on the c-level iso-curve at a time, and we need to sort and parameterize these points to obtain a real curve. All iso-curves are arc length parameterized as $\alpha(s), s \in [a, b]$. The initial point of the arc length parameterization locates on the symmetric plane of a face as shown in fig.3.1 (c), where the start points are marked as '*'. Arc length (s) is set to be negative on the left half and position on the right half. Therefore, the absolute value $|a|$ and $|b|$ is the arc length of left and right half of the iso-curve, and $b - a$ is the total arc length. Because of the diversity of arc length of iso-curves, ranges $[a, b]$ might vary on different images and we only focus on the common range shared by all images. In simulations, iso-curves are discretized as

$$P[k] = P(k\Delta l), \quad k = [a, b], \quad \Delta l = 2. \tag{3.9}$$

3.4 Discriminant Analysis

Because of facial expressions, faces are subject to non-rigid transformation, and some part of the iso-curves are also subject to this non-rigid transformation, especially near the mouth and cheeks. Thus, only being invariant to Euclidean transform is insufficient and we need to extract features that are also robust to facial expressions. For this purpose, discriminant analysis is applied to evaluate the robustness of EAF features.

Samples of EAF is denoted as $\theta_i[k] = \theta_i(k\Delta s)$, where i is the index of an iso-curve and Δs the sampling interval. The within class scatter of EAF is defined as $A_i[k] = E\{\theta_i[k]|\omega\}$ and the between class scatter is $B_i[k] = E\{\theta_i[k]\}$. If the training

set contains N subjects and M images for each subject, EAF of the m-th image of the n-th subject is denoted as $\{\theta_i^{m,n}[k]\}$. These two scatters can be computed as

$$A_i[k] = \frac{1}{M}\sum_{m=1}^{M}(\bar{\theta}_m^i[k] - \bar{\theta}^i[k])^2 \tag{3.10}$$

$$B_i[k] = \frac{1}{MN}\sum_{m=1}^{M}(\theta_{m,n}^i[k] - \bar{\theta}_m^i[k])^2 \tag{3.11}$$

where

$$\bar{\theta}_m^i[k] = \frac{1}{N}\sum_{n=1}^{N}\theta_{m,n}^i[k], \quad \bar{\theta}^i[k] = \frac{1}{MN}\sum_{n=1}^{N}\sum_{m=1}^{M}\theta_{m,n}^i[k]$$

We subsequently use the same separability confidence as in [25]

$$C_i[k] = \frac{A_i[k]}{B_i[k]} \tag{3.12}$$

It is obviously that the larger $C_i[k]$ is, the better separability $\theta_i[k]$ has. After obtaining the confidence, a weighting function is defined as Eq. 3.13, which thresholds the criteria $C^i[k]$, and only selects those features with $C_i[k] \geq \gamma$

$$W_i[k] = \begin{cases} 0, & \text{for } C_i[k] < \gamma \\ 1, & \text{for } C_i[k] \geq \gamma \end{cases} \tag{3.13}$$

3.4.1 Recognition

Face recognition has two scenarios, classification and verification. In face classification, we have a face gallery with n different subjects, and given a new input image, our target is to classify this input to one of n classes in the gallery. In other words, it is a n-class classification problem. Face verification is a hypothesis test problem that given two facial images, the goal is to make decision whether they belong to the same subject. Thus, face verification is a 2-class classification problem.

With weighting function, the matching distance between two EAFs, $\theta^i[k]$ and

$\tilde{\theta}^i[k]$, is defined as Equation. 3.14:

$$D(\theta, \tilde{\theta}) = \sum_{i=1}^{40} \sum_{k=a_i}^{b_i} \left[(\theta^i[k] - \tilde{\theta}^i[k]) \cdot w^i[k] \right]^2 \tag{3.14}$$

where, $[a_i, b_i]$ is the range of the i-th iso-curve. In the next section, the Receiver Operation Characteristic (ROC) curve based on this distance measure will be shown.

3.5 Experimental Results

We collected 30 subjects as a training set from Face Recognition Grand Challenge II (FRGC2) Spring 2004 database [37]. In the training set, each subject provides 4 different expressions. Our experiment is performed on 200 images of 50 subjects. For each subject, we collected two neutral images and two non-neutral images (including smile ,surprise, inflated and frown). First, 50 neutral faces are selected as gallery templates, and 150 independent 3D scans for testing. The rank1 and rank2 recognition rate based on the matching distance is provided in Table.3.1. The matching distance between every pair of images (19900 in total) is computed to generate Receiver Operation Characteristic (ROC) Curve, which is shown in Fig. 3.6.

Table 3.1: Rank1 and Rank2 recognition rate

	Neutral faces	Non-neutral faces
Rank1	100%	93.64%
Rank2	100%	95.46%

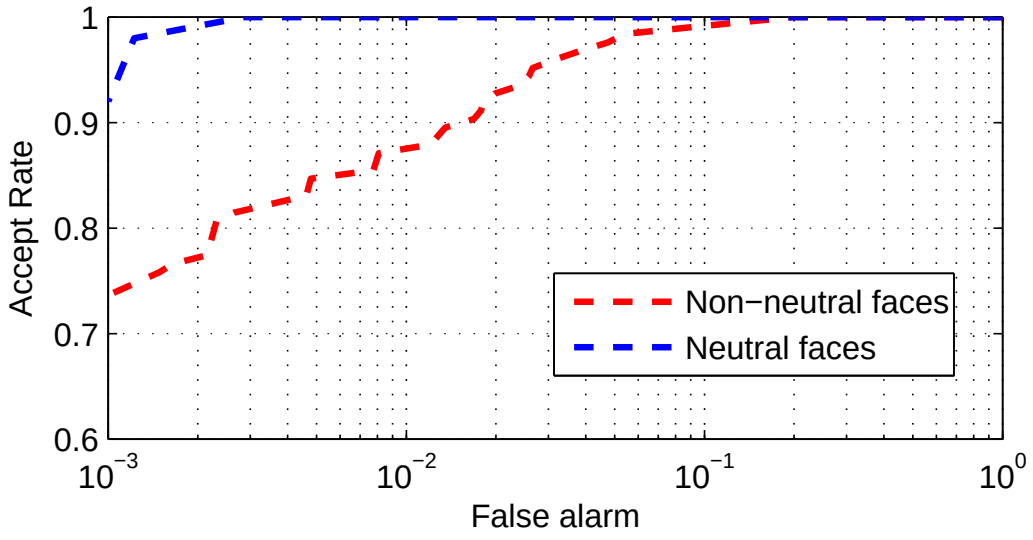


Figure 3.6: ROC curves for experiment on FRGC ver2.0 database.

Chapter 4

3D Surface Segmentation

Intuitively, facial expressions only affect part of a human face, and the part remaining invariant under expressions can be used for face recognition. In chapter 3, we extracted a set of facial curves and performed Discriminant Analysis to select expression-invariant part of these curves. In this chapter, we introduce a 3D surface segmentation scheme, which can adaptively extract identical parts between two facial images. Let's consider two cases: the first case is that these two images are from the same subject, either of same facial expressions or different, and the second case is that the two images are from different subjects. In the first case, the extracted information is the expression-invariant part between these two facial expressions. In the second case, because it is unlikely that faces of two different subjects can share a large identical area, we have the intuition that the partial similarity between them is small. Based on this argument, the identical facial area can be used as a distance measure between two facial images.

The remainder of the chapter is organized as follows, in Section 4.1, an rigid surface registration technique, the Iterative Closest Point (ICP) algorithm, is described as a background. In Section 4.2, an improvement on ICP algorithm is proposed to enable it to register two surfaces with non-rigid transformation. Experimental results and analysis are discussed in Section 4.3.

4.1 Iterative Closest Point

A major problem of utilizing 3D data is its subjection to Euclidean transform including rotation and translation. The ICP algorithm [6][7] is a technique to register two 3D surfaces by estimating the best rigid transform between them, which minimize the mean square error (MSE). With an initial estimation of a rigid transform, the ICP algorithm iteratively chooses the corresponding point and refines the rotation R and translation T to minimize the mean square distance.

4.1.1 Point Set Registration

A major part of ICP algorithm is the closest point registration. Given two point sets A and B, according to the closest point registration, any point in A is correspondent to the point in B with the smallest Euclidean distance to it. The mean square distance between these two point sets is defined as the average of the square Euclidean distance of all correspondent points. The Euclidean distance $d(\vec{r_1}, \vec{r_2})$ between two points $\vec{r_1} = [x_1, y_1, z_1]^T$ and $\vec{r_2} = [x_2, y_2, z_2]^T$ is $d(\vec{r_1}, \vec{r_2}) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. For two point sets $A = \{\vec{a_i}\}$ and $B = \{\vec{b_j}\}$, $i = 1, ..., N_a$ and $j = 1, ..., N_b$, the index of the correspondent point of $a \in A$ is

$$\phi(i) = arg \min_{i \in \{1,...,N_b\}, \vec{b_i} \in B} d(\vec{a}, \vec{b_i}) \tag{4.1}$$

Based on the closest point registration, the mean square distance between two point sets is defined as

$$d(A, B) = \frac{1}{N_a} \sum_{i=1}^{N_a} d(\vec{a_i}, \vec{b_{\phi(i)}})^2 \tag{4.2}$$

where $\phi(i)$ is the index of the correspondent point of $\vec{a_i}$. Without losing generality, in the following sections, we denote $\vec{b_i}$ as the closest point to the point $\vec{a_i}$.

4.1.2 Optimal Rigid Transformation

After registering two point sets, the ICP algorithm solves the optimal rigid transformation by iteratively minimizing the mean square distance as shown in Equation 4.2. A well known representation of rotation in R^3 is a 3×3 rotation matrix, which can be decomposed into three independent rotations in three directions,

$$R(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

The translation is represented by a vector $T = [x, y, z]^T$. Based on the rotation matrix and the translation vector, the Euclidean transformation of a point x is: $x' = Rx + T$. The sum of squares of distance becomes

$$f(R, T) = \sum_{i=1}^{N} ||a_i - Rb_i - T||^2 \tag{4.4}$$

or

$$f(R, T) = \sum_{i=1}^{N} ||a_i - Rb_i||^2 - 2T \cdot \sum_{i=1}^{N} [a_i - Rb_i] + N||T||^2 \tag{4.5}$$

The total error is obviously minimized by

$$T = \frac{1}{N} \sum_{i=1}^{N} [a_i - Rb_i] = \mu_a - R\mu_b, \tag{4.6}$$

where

$$\mu_a = \frac{1}{N} \sum_{i=1}^{N} a_i, \qquad \mu_b = \frac{1}{N} \sum_{i=1}^{N} b_i. \tag{4.7}$$

The translation is just the difference of the centroid of the point set B and the centroid of the point set A. Accordingly, by moving the centroid of each point set to the original of the coordinates, we can also minimize Equation 4.4. Substituting

$a'_i = a_i - \mu_a$ and $b'_i = b_i - \mu_b$ into Equation 4.4, it becomes

$$f(R,T) = \sum_{i=1}^{N} ||a'_i - Rb'_i||^2. \tag{4.8}$$

Expand the total error, we obtain

$$f(R,T) = \sum_{i=1}^{N} ||a'_i||^2 + 2\sum_{i=1}^{N} a'_i \cdot Rb'_i + \sum_{i=1}^{N} ||Rb'_i||^2 \tag{4.9}$$

Because

$$||Rb'_i||^2 = ||b'_i||^2, \tag{4.10}$$

minimizing the total error is equivalent to maximizing

$$\sum_{i=1}^{N} a'_i \cdot Rb'_i. \tag{4.11}$$

However, because the rotation matrix has sinusoid functions of 3 parameters, maximizing Equation 4.11 is usually very difficult. Horn *et al.* [38] proposed to represent the rotation by a unit quaternion $q_R = q_0 + iq_1 + jq_2 + kq_3$, and provided a closed form solution of Equation 4.11. His method is described as follows.

A quaternion can be thought as a complex number with three imaginary pasts, which have the following properties,

$$i^2 = -1, \quad j^2 = -1, \quad k^2 = -1,$$
$$ij = -k, \quad jk = -i, \quad ik = -j$$

Then, the inner product of two quaternions is

$$\begin{aligned}\dot{r}\dot{q} &= (r_0q_o - r_1q_1 - r_2q_2 - r_3q_3) \\ &\quad +i(r_0q_1 + r_1q_0 + r_2q_3 - r_3q_2) \\ &\quad +j(r_0q_2 - r_1q_3 + r_2q_0 + r_3q_1) \\ &\quad +k(r_0q_3 + r_1q_2 - r_2q_1 + r_3q_0)\end{aligned}$$

which can also be represented in a matrix form,

$$\dot{r}\dot{q} = \begin{bmatrix} r_0 & -r_1 & -r_2 & -r_3 \\ r_1 & r_0 & -r_3 & r_2 \\ r_2 & r_3 & r_0 & -r_1 \\ r_3 & -r_2 & r_1 & r_0 \end{bmatrix} \dot{q} = \Re\dot{q}.$$

and

$$\dot{q}\dot{r} = \begin{bmatrix} r_0 & -r_1 & -r_2 & -r_3 \\ r_1 & r_0 & r_3 & -r_2 \\ r_2 & -r_3 & r_0 & r_1 \\ r_3 & r_2 & -r_1 & r_0 \end{bmatrix} \dot{q} = \bar{\Re}\dot{q}.$$

A point (x, y, z) in R^3 is represented by a quaternion $\dot{r} = ix + jy + kz$, whose real part is zero. The rotation of a point $\dot{r}$ can be represented by

$$\dot{r'} = \dot{q}\dot{r}\dot{q}* = (\bar{Q}^T Q)\dot{r}, \tag{4.12}$$

where $\dot{q}$ is a quaternion with $q_0 > 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. Then, the rotation

matrix is

$$R(\vec{q_R}) = \bar{Q}^T Q = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \tag{4.13}$$

This representation of the rotation matrix has a polynomial form, which makes the optimization problem much easier. Now, substituting Equation 4.12 to Equation 4.11, the target function needed to be maximized becomes

$$\sum_{i=1}^{N} \dot{a_i'} \cdot (\dot{q}\dot{b_i'}\dot{q}*). \tag{4.14}$$

This formula cam be rewritten as

$$\sum_{i=1}^{N} (\dot{q}\dot{a_i'}) \cdot (\dot{b_i'}\dot{q}), \tag{4.15}$$

which is

$$\dot{q}^T \left(\sum_{i=1}^{N} \bar{A}_i^T B_i \right) \dot{q}, \tag{4.16}$$

where A_i and B_i is the corresponding matrix of point a_i and b_i. Now, it is obvious that Equation 4.11 is maximized by choosing $\dot{q}$ as the eigenvector corresponding to the most positive eigenvalue of the matrix $\left(\sum_{i=1}^{N} \bar{A}_i^T B_i \right)$.

A 3×3 anti-symmetric matrix is defined as $M = (\sum_{ab} - \sum_{ab}^T)$, where $\sum_{ab} = \sum_{i=1}^{N} a_i' b_i'$, and the cyclic components of matrix M are used to form a column vector $\Delta = [M_{23} \quad M_{31} \quad M_{12}]^T$. This vector is used to compute the matrix $\left(\sum_{i=1}^{N} \bar{A}_i^T B_i \right)$.

$$\left(\sum_{i=1}^{N} \bar{A}_i^T B_i \right) = Q(\sum_{ab}) = \begin{bmatrix} tr(\sum_{ab}) & \Delta^T \\ \Delta & \sum_{ab} + \sum_{ab}^T - tr(\sum_{ab}) I_3 \end{bmatrix} \tag{4.17}$$

In the end, the optimal rotation vector is obtained by computing the unit eigenvec-

tor $\vec{q_R} = [q_0 \quad q_1 \quad q_2 \quad q_3]^T$ corresponding to the maximum eigenvalue of the matrix $Q(\sum_{ab})$ and the optimal translation vector is given by $\vec{q_T} = \vec{\mu_b} - R(\vec{q_R})\vec{\mu_a}$. (For explicit proof, please see the reference [38])

4.1.3 Iterative Registration Algorithm

Given the point set A with N_a points from a data shape, and the point set B with N_b points from a model shape, the ICP algorithm can be stated as follows. First of all, the initial registration vector $\vec{p_0}$, named coarse alignment, is obtained by Principle Component Analysis (PCA). For both the data shape and the model shape, three eigenvectors corresponding to three largest eigenvalues are extracted, and for each shape, there is a reference plane spanned by the first two eigenvectors. The coarse alignment is a rigid transform, which can make these two reference planes overlap, and leave the third eigenvector on the same side of the plane. After the coarse alignment, ICP algorithm is performed as follows.

1. Perform coarse alignment.
2. Obtain correspondent pairs (a_i, b_i) by closest point registration.
3. Estimate the registration vector by choosing $\vec{q_R}$ as the eigenvector corresponding to the largest eigenvalue of the matrix defined in Equation 4.17, and $\vec{q_T} = \vec{\mu_b} - R(\vec{q_R})\vec{\mu_a}$.
4. Compute the mean square distance D_k.
5. Terminate the iteration if $||D_k - D_{k-1}|| < \lambda$, otherwise go back to step 2

It can be proved that the ICP algorithm can always converge to a local minimum of the mean square distance target function. To prove this, let e_k denote the mean square distance in the k-th iteration before performing Euclidean transformation, and f_k denote the mean square distance after performing Euclidean transformation. Because the optimal Euclidean transformation is to minimize the means square distance, we know that $e_k > f_k$. Because e_{k+1} is obtained by performing closest point registration,

obviously we have $f_k > e_{k+1}$. Therefore, the mean square distance is monotonically decreasing as the algorithm iterates

$$e_0 > f_0 > ... > e_k > f_k > e_{k+1} > f_{k+1} > ... \tag{4.18}$$

Thus, the ICP algorithm always converges to a local minimum.

4.2 Improvement of the ICP Algorithm

Let's consider a very common example, which shows the limitation of the ICP algorithm. In Fig 4.1, we have two surfaces that are partially identical except for a pyramid on the red surface. Because ICP uses the mean square distance as its target function, it will definitely lead to a misalignment when there is a non-rigid deformation, as shown in Fig 4.2. However, in our face recognition applications, due to the variation of facial expressions, human faces are usually subjected to non-rigid deformation, which will severely undermine the performance of the ICP registration. In this section, an improvement of ICP is introduced to partially register non-rigid deformed surfaces.

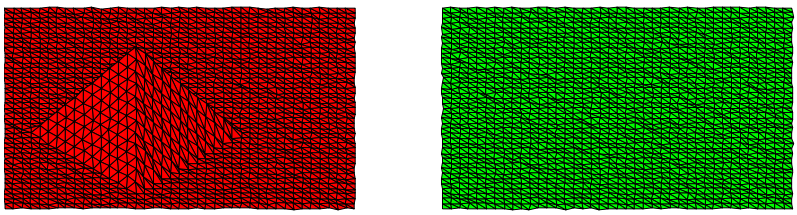


Figure 4.1: Two surfaces with partial similarity

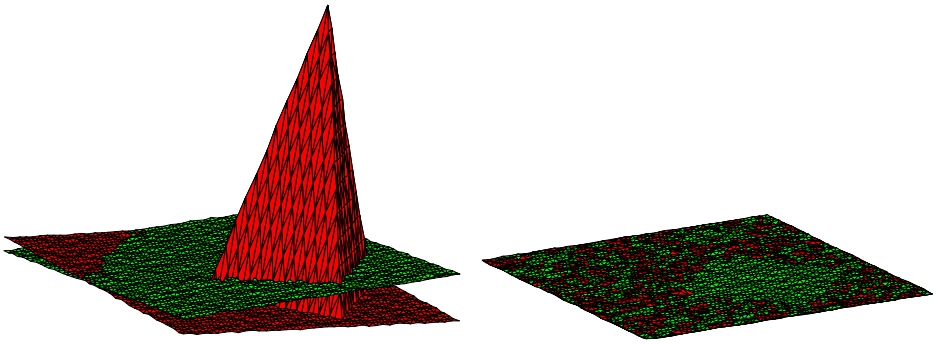
If we watch human faces under different facial expressions carefully, we will notice that under most facial expressions, the invariant area of human faces is usually larger than the deformed area. Therefore, in this section, we make the assumption that a major part of the facial area remains invariant to facial expressions. Admittedly, there are certain extreme facial expressions conflicting with the assumption, on which the

surface registration algorithm proposed in this section will fail. However, making ICP be able to register 3D faces under most facial expressions is also a very meaningful improvement.

Given two 3D face images of the same subject, it is the deformed areas of these two faces that undermine the ICP registration by introducing a misalignment, which cannot be canceled by a rigid transformation. Thus, the ICP algorithm will be able to provide a good surface registration if the non-rigid deformed area is removed from the surface. Intuitively, if the majority of these two triangular surface are similar, the nodes contributing the largest error belong to the deformed area. Based on this argument, we make an improvement on the the ICP algorithm by removing the most significant error contributor iteratively until the mean square distance is reduced to a pre-defined threshold λ. The improved ICP algorithm is illustrated as follows,

1. Estimate a registration vector $\vec{p} = [\vec{p_R} \quad \vec{p_T}]^T$ by the ICP algorithm.
2. Compute the square error contributed by each node b_i in B, $C(b_i) = d(A, R(\vec{q_R})b_i + \vec{q_T})$.
3. Remove b_i with the largest value $C(b_i)$ from the point set B.
4. Estimate the registration vector by choosing $\vec{q_R}$ as the eigenvector corresponding to the largest eigenvalue of the matrix defined in Equation 4.17, and $\vec{q_T} = \vec{\mu_b} - R(\vec{q_R})\vec{\mu_a}$.
5. Update $C(b_i)$ for each node. Compute the mean square distance as $D_k = \sum_{i=1}^{N_b} C(b_i)$.
6. Stop iteration if $||D_k - D_{k-1}|| < \mu$, if not go back to step 3.

In Fig 4.2, an experiment on the previous example shows that the improved ICP algorithm successfully removes the deformed part and significantly enhances the precision of the surface registration. Experiments performed on 3D face data also show that the face area remaining invariant under the presented facial expression are successfully extracted by the improved ICP algorithm.



(a) (b)

Figure 4.2: (a) ICP registration (b) Improved ICP registration

4.3 Recognition and Experimental Results

The improved ICP algorithm segments identical areas between two facial images, and the area will be used for recognition. As illustrated before, it is unlikely that there are large identical areas between two different subjects, while images from the same subject usually exhibits considerably large identical areas even when they are captured under different facial expressions. Based on this observation, we simply use the size of matched area as the distance measure between facial images, and the nearest neighbor classifier is applied for classification.

In the experiment, we use the same data set as in Section 3.5, which consists of 50 subjects and 4 images for each, including 2 neutral faces and 2 non-neutral faces. Because the improved ICP algorithm dose not require training procedure, we directly compute the distance measure to obtain accept rates and false alarm rates with different decision boundary. A ROC curve of the presented 3D face recognition technique is shown in Fig 4.4. Comparing with the experimental results shown in Section 3.5, we can clearly see an significant improvement of the recognition rate under the variability of facial expressions.

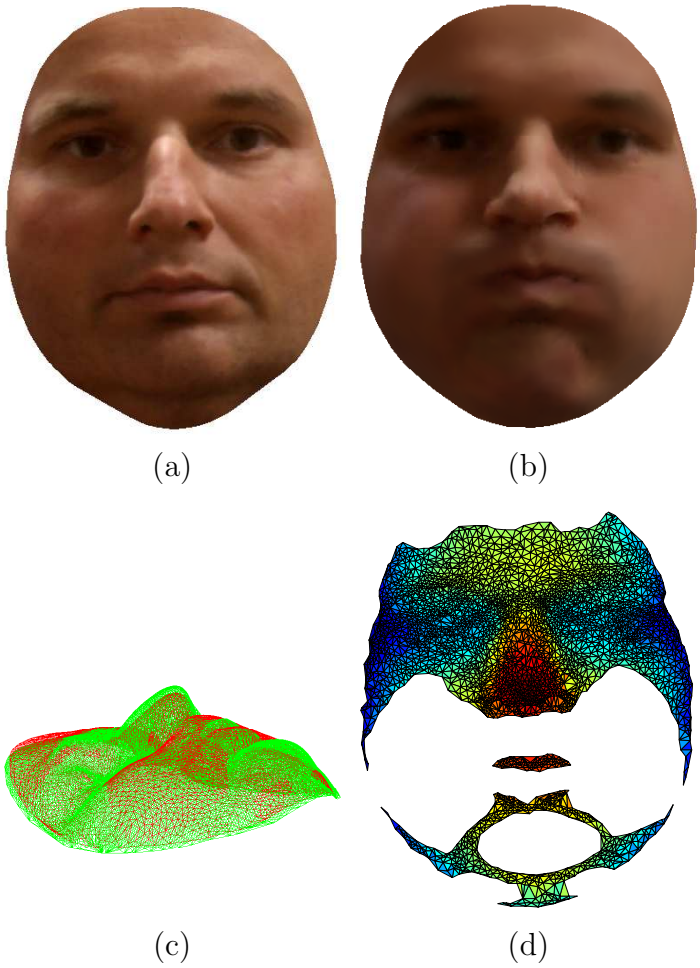


Figure 4.3: (a) Face 1 (b) Face 2 (c) Rigid surface matching between the two image (d) Segmented partial similar area

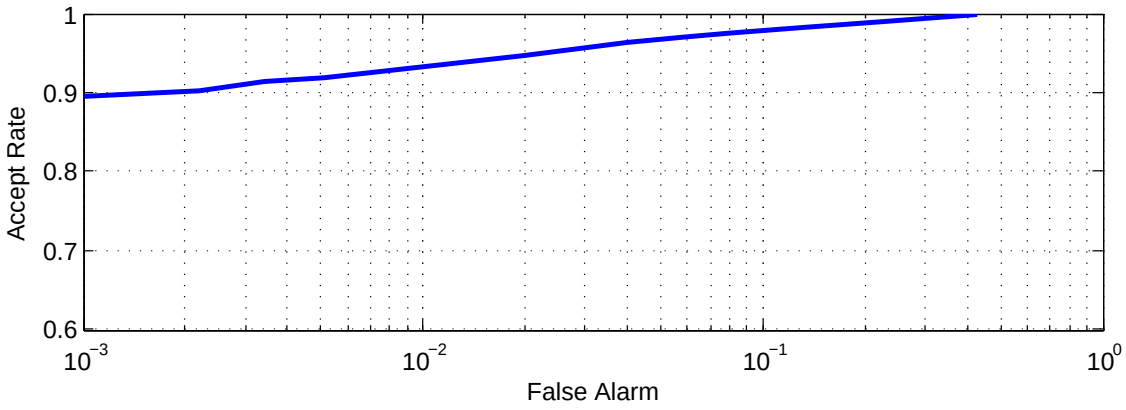


Figure 4.4: ROC for face recognition based on improved ICP algorithm

Chapter 5

Future Work in 3D Segmentation by the Level Set Method

As stated before, the improved ICP algorithm is based on an assumption that the majority of a face area will remain invariant under most facial expressions. When certain extreme facial expressions violate this assumption, the improved ICP algorithm might fail. In this chapter, we propose to use level set method for detecting partial similarity between facial images. Given two 3D facial images, the general idea is to grow a level set curve on one given facial image, with the constrain that the facial areas inside the level set curve is identical to a subset of the other image. The initialization of the level set curve is a rectangular around the nose tip, which is expression-insensitive even under extreme facial expressions. Then the level curve evolves on the face to absorb more area, which is identical to the other image. The evolution is guided by an Eikonal partial differential equation, and solved by a fast marching method.

The remainder of the chapter is organized as follows. In Section 5.1, the basic concepts of level set method and fast marching method are introduced. In Section 5.2, I formalize the 3D segmentation problem to be a problem of evolving a level set curve, and provide the framework of a novel 3D segmentation scheme based on the level set method. Future works about 3D segmentation by the level set method is

discussed in Section 5.3.

5.1 Background

5.1.1 Level Set Method

Level set method was first proposed by Osher and Sethian [39][35]. In this subsection, some of the basic concepts and implementation details of level set method are introduced. A level set curve defined by a level set function $\phi(x, y, t)$ is a path $(x(t), y(t))$ on which the level set value must be zero. Hence,

$$\phi(x(t), y(t), t) = 0. \tag{5.1}$$

By the chain rule,

$$\phi_t + \nabla\phi(x, y) \cdot \vec{v}(x, y) = 0, \tag{5.2}$$

where $\vec{v}(x, y) = [x'(t), y'(t)]$. If F supplies the speed in the outward normal direction, then $F = \vec{v} \cdot \vec{n}$, where $\vec{n} = \nabla\phi/|\nabla\phi|$. The evolution equation for ϕ is

$$\phi_t + F|\nabla\phi| = 0. \tag{5.3}$$

Given an interface identified as a level set curve of the level set function ϕ, Equation 5.3 describes the time evolution of the interface. Consider a special and simple case where the level set function $\phi(x, y)$ is defined as the time at which the interface crosses the point (x, y), denoted as $T(x, y)$. In this special case, the derivative of the level set function with respect to time t is always equal to 1, thus Equation 5.3 becomes a partial differential equation,

$$|\nabla T|F = 1. \tag{5.4}$$

This is a form of the well-known Eikonal equation, which can generally be solved by a fast marching method [35]. The boundary condition in this case is $T(x, y) = 0$ on

the initialization curve.

Because usually we need to solve the eikonal equation on a image plane, where the domain is divided into grid points sampled in both x and y direction, it's necessary to construct an accurate approximation of the gradients ∇T. An upwind scheme given in [40] is shown to be a very efficient and precise approximation of the gradients. The level set function $T(x, y)$ can be expanded as a Taylor series in both x and y direction, thus we have

$$T(x+\Delta, y) = T(x, y) + \frac{\partial T(x, y)}{\partial x}\Delta + O(\Delta^2)$$
$$T(x, y+\Delta) = T(x, y) + \frac{\partial T(x, y)}{\partial y}\Delta + O(\Delta^2)$$

After rearranging this expression, the derivative at point (x, y) can be written as

$$T_x(x, y) = \frac{T(x+\Delta, y) - T(x, y)}{\Delta} + O(\Delta)$$

Then, differential operators are defined as

$$\begin{aligned} D^{+x}T &= \frac{T(x+\Delta, y) - T(x, y)}{\Delta}, \\ D^{-x}T &= \frac{T(x, y) - T(x-\Delta, y)}{\Delta}, \\ D^{0x}T &= \frac{T(x+\Delta, y) - T(x-\Delta, y)}{2\Delta}. \end{aligned} \tag{5.5}$$

According to the upwind scheme, only values upwind of the direction of information propagation should be used. Therefore, the gradient is approximated by

$$\begin{aligned} |\nabla T| &= [\max(D_i^{-x}j, 0)^2 + \min(D_i^{+x}j, 0)^2, \\ &\quad + \max(D_i^{-y}j, 0)^2 + \min(D_i^{+y}j, 0)^2]^{1/2}. \end{aligned} \tag{5.6}$$

Substitute Equation 5.6 into Equation 5.4, we can obtain,

$$\begin{aligned}\frac{1}{F^2} &= \max(D_i^{-x}j,0)^2 + \min(D_i^{+x}j,0)^2, \\ &\quad + \max(D_i^{-y}j,0)^2 + \min(D_i^{+y}j,0)^2. \end{aligned} \tag{5.7}$$

A similar approximation to the gradient was introduced by Rouy and Tourin [41],

$$\begin{aligned}\frac{1}{F^2} &= \max\big(\max(D_i^{-x}j,0)^2, -\min(D_i^{+x}j,0)^2\big), \\ &\quad + \max\big(\max(D_i^{-y}j,0)^2, -\min(D_i^{+y}j,0)^2\big). \end{aligned} \tag{5.8}$$

5.1.2 Fast Marching Method

A very efficient numerical solution of the Eikonal equation, named fast marching algorithm was proposed by Sethian [35]. For this discussion, we limit ourselves to a positive speed function $F(x,y)$, which makes the level set curve monotonically propagate off the initial line.

The key of fast marching algorithm is that the upwind different structure means the information propagates "one way", from smaller values of $T(x,y)$ to larger ones, which meets the assumption of a positive speed function. Because the idea of fast marching method is to update the value $T(x,y)$ around the existing front iteratively, the selection of which grid to update around the front is the key of the whole algorithm. Because of the assumption that the front propagates "one way", the smallest value around the front is always correct; other points around the front with larger $T(x,y)$ cannot affect it. The technique is explained algorithmically as follows,

1. Definition

   (a) Alive points: points whose value of $T(x,y)$ is known for sure. Letting (i,j) denote the index of grid, the set of alive points is represented by $\Omega = \{(i,j)\}$.

   (b) Narrow band points: points which are not alive points but adjacent to at least one alive point. (For the grid (i,j), the adjacent grids are $(i-1,j)$,

$(i+1, j)$, $(i, j-1)$, $(i, j+1)$). The value of T at the *Narrow Band* points are estimated by Equation 5.8.

(c) Far away points: points which are neither alive points nor narrow band points. The value of T at the *Far Away* points are set to be infinity.

2. Fast Marching Algorithm

(a) Begin Loop: Let (i_{min}, j_{min}) be the point in *Narrow Band* with the smallest value for T.

(b) Add the point (i_{min}, j_{min}) to the Ω, and remove it from *Narrow Band*.

(c) Tag as neighbors any points $(i_{min}-1, j_{min})$, $(i_{min}+1, j_{min})$, $(i_{min}, j_{min}-1)$, $(i_{min}, j_{min}+1)$ that are either in *Narrow Band* or *Far Away*. All these neighbors are them moved to the *Narrow Band* set.

(d) Recompute the values of T at all neighbors according to the largest possible solution to the quadratic equation (Equation 5.8).

(e) Return to the top of the loop

Because the recomputed T is selected to be that largest possible solution to the quadratic equation, whenever converting a *Narrow Band* point to *Alive*, the process of recomputing the T value of its neighbors cannot produce a value less than any *Alive* value. This property guarantee that the level set curve is always marching ahead in time.

The discrete form of Equation 5.8 is given by

$$\begin{aligned} f_{ij}^2 &= \max\big(\max(D_i^{-x}j, 0)^2, -\min(D_i^{+x}j, 0)^2\big), \\ &\quad + \max\big(\max(D_i^{-y}j, 0)^2, -\min(D_i^{+y}j, 0)^2\big), \end{aligned} \tag{5.9}$$

where the inverse of speed function F_{ij} is replaced by $f_{ij} = \frac{1}{F^2}$. Consider a the matrix of grid values in Fig. 5.1 from [35]. The goal is to compute the new value of T in the center grid to update it, based on its neighbors A, B, C, D. Without loosing

	B	
A	?	C
	D	

Figure 5.1: Matrix of adjacent grids

generality, the value A is assumed to be the smallest value of T in the *Narrow Band* set. The problem can be generally divided into to cases.

case 1: C is not alive. In this case, the T value of the center grid should be computed from A.

1. If $A + f \leq \min(B, D)$, $T_{computed} = (A + f)$.

2. If $A + f > \min(B, D)$, with out loosing generality, we assume that $B <= D$. The value of T is obtained by solving the quadratic equation $(T_{computed} - A)^2 + (T_{computed} - B)^2 = f^2$

case 2: C is alive. In this case, the T value of the center grid should be computed from C. This case defaults into the first case above.

5.2 Surface Segmentation by Solving Eikonal Equation

As discussed above, the algorithm introduced in Section 4.2 cannot cope with extreme facial expressions, as the invariance of the majority of facial area is not valid. Our future work will be focused on the surface segmentation scheme based on the level set method, which is much more robust under extreme facial expressions.

The facial range images we have are represented as six matrix $\boldsymbol{(X, Y, Z, R, G, B)}$, which provide us a depth image $z = f(x, y)$. However, the fast marching algorithm cannot be directly applied, as 3D images captured by most 3D scanners are usually

not uniform sampled. Thus, as a preprocessing procedure, $(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})$ is uniformly sampled to generate depth image $z[i,j] = f(i\Delta, j\Delta)$, where i, j denotes the discrete x, y coordinates.

By the depth images, the 3D segmentation can be performed by segmenting the parameter space of (x, y), which can be formalized by a level set method. The segmentation of the a 3D face is represented by a level set function $T(x, y)$. Given a level value c, the surface is segmented into two parts: $S_{in}(c) = \{(x, y)|T(x, y) \leq c\}$ and $S_{out}(c) = \{(x, y)|T(x, y) > c\}$, where S_{in} is the region inside the c-level set curve, and S_{out} is the region outside it. In this section, we select appropriate speed vector in order to make S_{in} to be the facial region which is invariant to the presented facial expression, and S_{out} to be the region which undergoes a non-rigid deformation.

To solve the Eikonal equation, a boundary condition is needed. As shown by Chang, K.I. *et al.* [42], the nose region is the most stable part of human face which remains invariant under extreme facial expressions. In other words, given two 3D facial images A and B of the same subject, a small area around the nose tip in A can be perfectly matched to B. Therefore, the initial curve is chosen to be a small rectangular near the nose tip in A, and the value $T(x, y)$ inside the initial curve is set to be 0. Based on the above argument, we know that if the ICP algorithm is performed on $S_{in}(0)$ and B, the mean square distance between them should be almost zero. Obviously, a zero mean square distance between two 3D surfaces indicates that these two surfaces are identical. The basic idea is to grow the region S_{in} with the constraint that the mean square distance between S_{in} and B is close to zero, which indicates that they are identical. Intuitively, we would like the front of the level set curve to propagate faster in those regions which are invariant to the presented facial expression, and slower in those regions which are affected by a non-rigid deformation. Thus, I propose to make the speed vector to be $F(x, y) = \frac{1}{D(x,y)^{\lambda}}$, where $D(x, y)$ is the distance to surface B based on closest point registration. This is a monotonically decreasing function of $D(x, y)$, and it satisfies the intuition that a higher speed should be assigned to the regions which are more similar, and lower speed to the regions which are more dissimilar. The 3D segmentation scheme is explained algorithmically

as follows,

1. Detect a rectangular region near nose tip, $S_{in}(0)$,as the boundary condition of the Eikonal equation. According to fast marching algorithm, S_{in} is the *Alive* set. Based on this, initialize the *Narrow Band* and the *Far Away* sets.

2. Beginning of the loop: Estimate registration vector $\vec{p} = [\vec{p_R} \quad \vec{p_T}]^T$ between S_{in} and B by the ICP algorithm. Perform the rotation $R(\vec{p_R})$ and the translation $\vec{p_T}$ on the surface A.

3. Compute the distance from each node $\{x, y, z = f(x, y)\}$ in A to B, denoted as $D(x, y)$.

4. Set $F(x, y)$ to be $F(x, y) = \frac{1}{D(x,y)^\lambda}$. And $f(x, y) = D(x, y)^\lambda$.

5. Let (i_{min}, j_{min}) be the point in *Narrow Band* with the smallest value for T. Add the point (i_{min}, j_{min}) to the S_{in}, and remove it from *Narrow Band*.

6. Tag as neighbors any points $(i_{min} - 1, j_{min})$, $(i_{min} + 1, j_{min})$, $(i_{min}, j_{min} - 1)$, $(i_{min}, j_{min} + 1)$ that are either in *Narrow Band* or *Far Away*. All these neighbors are them moved to the *Narrow Band* set.

7. Recompute the values of T at all neighbors according to the largest possible solution to the quadratic equation (Equation 5.8).

8. Return to the top of the loop

5.3 Discussion and Future Work

The form of the speed function $F(x, y)$ has very a significant impact on the evolution result. Thus, one part of the future work is the selection of the speed function $F(x, y)$. As discussed before, the front of the level set curve should propagate faster in regions where two surfaces are identical, and propagate slower or even stop in regions where they are different. In the previous section, the Euclidean distance $D(x, y)$ is

used as a measurement of the surface similarity, and the speed function $F(x, y)$ is set to be a monotonically decreasing function of $D(x, y)$. However, some more complex similarity measurements, like curvature or shape index, might be better. Thus, it is necessary to analyze the performance of these similarity measurements, as well as other forms of the speed function $F(x, y)$.

Reducing the computational cost of the algorithm is also a key part of the future work. Because the fast marching method is an iterative algorithm, which requires surface registration in each iteration, it is usually quite heavy in computation. Thus, in the future I will analyze the complexity of the algorithm and make improvements on the efficiency.

Bibliography

[1] Andrea F. Abate, Michele Nappi, Daniel Riccio, and Gabriele Sabatino. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885 – 1906, 2007.

[2] Xiaoguang Lu, Rein-Lien Hsu, A.K. Jain, B. Kamgar-Parsi, and B. Kamgar-Parsi. Face recognition with 3d model-based synthesis. In *Biometric Authentication. First International Conference, ICBA 2004. Proceedings (Lecture Notes in Comput. Sci. Vol.3072)*, pages 139–46, 2004.

[3] I. Mpiperis, S. Malassiotis, and M.G. Strintzis. 3-d face recognition with the geodesic polar representation. *Information Forensics and Security, IEEE Transactions on*, 2(3):537–547, Sept. 2007.

[4] F. Tsalakanidou, D. Tzovaras, and M. G. Strintzis. Use of depth and colour eigenfaces for face recognition. *Pattern Recognition Letters*, 24(9-10):1427 – 1435, 2003.

[5] Bernard Achermann, Xiaoyi Jiang, and Horst Bunke. Face recognition using range images. *Virtual Systems and MultiMedia, International Conference on*, 0:129, 1997.

[6] P.J. Besl and H.D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb 1992.

[7] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119 – 52, 1994.

[8] D. Colbry, G. Stockman, and Anil Jain. Detection of anchor points for 3d face veri.cation. In *Computer Vision and Pattern Recognition - Workshops. IEEE Computer Society Conference on*, pages 118–118, June 2005.

[9] C. Dorai and A.K. Jain. Cosmos-a representation scheme for 3d free-form objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(10):1115–1130, Oct 1997.

[10] G. Gordon. Face recognition based on depth maps and surface curvature. *Geometric Methods in Computer Vision, SPIE*, 1991.

[11] A.B. Moreno, A. S?chez, E. Fr?s-Mart?ez, and J.F. V?ez. Three-dimensional facial surface modeling applied to recognition. *Engineering Applications of Artificial Intelligence*, 22(8):1233 – 1244, 2009.

[12] Alessandro Colombo, Claudio Cusano, and Raimondo Schettini. 3d face detection using curvature analysis. *Pattern Recognition*, 39(3):444–455, 2006.

[13] Xiaoguang Lu and A.K. Jain. Deformation modeling for robust 3d face matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(8):1346–1357, Aug. 2008.

[14] C. Samir, A. Srivastava, and M. Daoudi. Three-dimensional face recognition using shapes of facial curves. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1858–1863, Nov. 2006.

[15] S. Feng, H. Krim, and I. A. Kogan. 3d face recognition using euclidean integral invariants signature. In *Statistical Signal Processing, IEEE Signal Processing Workshop on*, pages 156–160, Aug. 2007.

[16] A. Srivastava, C. Samir, S. H. Joshi, and M. Daoudi. Elastic shape models for face analysis using curvilinear coordinates. *Journal of Mathematical Imaging and Vision*, pages 253–265, June 2008.

[17] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(9):1063–1074, Sept. 2003.

[18] T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, Jun 2001.

[19] G. Passalis, I.A. Kakadiaris, T. Theoharis, G. Toderici, and N. Murtuza. Evaluation of 3d face recognition in the presence of facial expressions: an annotated deformable model approach. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pages 171–171, June 2005.

[20] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. Removing shape-preserving transformations in square-root elastic (sre) framework for shape analysis of curves. In *Energy Minimization Methods in Computer Vision and Pattern Recognition. Proceedings 6th International Conference*, pages 388–98, 2007.

[21] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn. A novel representation for riemannian analysis of elastic curves in rn. In *Computer Vision and Pattern Recognition, IEEE Conference on*, pages 1–7, June 2007.

[22] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Expression-invariant representations of faces. *Image Processing, IEEE Transactions on*, 16(1):188–197, Jan. 2007.

[23] A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, pages 5–30, 2005.

[24] Xiaoxing Li and Hao Zhang. Adapting geometric attributes for expression-invariant 3d face recognition. In *Shape Modeling and Applications, IEEE International Conference on*, pages 21–32, June 2007.

[25] Xiaoxing Li, Tao Jia, and Hao Zhang. Expression-insensitive 3d face recognition using sparse representation. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 2575–2582, June 2009.

[26] Xiaoguang Lu and A.K. Jain. Deformation analysis for 3d face matching. In *Application of Computer Vision, IEEE Workshops on*, volume 1, pages 99–104, Jan. 2005.

[27] G.J. Edwards, T.F. Cootes, and C.J. Taylor. Face recognition using active appearance models. In *Computer Vision - ECCV'98. 5th European Conference on Computer Vision. Proceedings*, volume vol.2, pages 581–95, 1998.

[28] J. Huang, V. Blanz, and B. Heisele. Face recognition using component-based svm classification and morphable models. In *Proceedings of Pattern Recognition with Support Vector Machines. First International Workshop.*, pages 334–41, 2002.

[29] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. In *7th European Conference on Computer Vision. Proceedings, Part IV*, pages 3–19, 2002.

[30] V. Blanz, S. Romdhani, and T. Vetter. Face identification across different poses and illuminations with a 3d morphable model. In *Automatic Face and Gesture Recognition, IEEE International Conference on*, volume 0, page 0202, 2002.

[31] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

[32] J. Huang, B. Heisele, and V. Blanz. Component-based face recognition with 3d morphable models. In *Audio- and Video-Based Biometric Person Authentication. 4th International Conference, AVBPA 2003. Proceedings (Lecture Notes in Computer Science Vol.2688)*, pages 27–34, 2003.

[33] Chenghua Xu, Yunhong Wang, Tieniu Tan, and Long Quan. Automatic 3d face recognition combining global geometric features with local shape variation information. In *Proceedings of Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 308–13, 2004.

[34] I.A. Kakadiaris, G. Passalis, G. Toderici, M.N. Murtuza, Yunliang Lu, N. Karampatziakis, and T. Theoharis. Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4):640–649, April 2007.

[35] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, 1996.

[36] S. Feng, H. Krim, and I. A. Kogan. 3d face recognition using euclidean integral invariants signature. In *Statistical Signal Processing, 2007. SSP '07. IEEE/SP 14th Workshop on*, pages 156–160, Aug. 2007.

[37] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 947–954 vol. 1, June 2005.

[38] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A (Optics and Image Science)*, 4(4):629 – 42, 1987/04/.

[39] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[40] J. A. Sethian. Level set methods and fast marching methods - evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. In *Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1998.

[41] Elisabeth Rouy and Agns Tourin. A viscosity solutions approach to shape-from-shading. *SIAM Journal on Numerical Analysis*, 29(3):867–884, 1992.

[42] K.I. Chang, W. Bowyer, and P.J. Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(10):1695 –1700, oct. 2006.