Three Dimensional Face Surfaces Analysis using Geodesic Distance

Rachid AHDID*, Said SAFI, Bouzid MANAUT
Interdisciplinary Laboratory of Research in Sciences and Technologies (LIRST), Sultan Moulay Slimane University, Beni Mellal, Morocco
*Corresponding author: r.ahdid@usms.ma
Received March 04, 2015; Revised April 16, 2015; Accepted May 06, 2015

Abstract In this paper, we present an automatic 3D face recognition system based on the computation of the geodesic distance between the reference point and the other points in the 3D face surface. To compute a geodesic distance, we use the Fast Marching algorithm for solving the Eikonal equation. For space reduction, we apply Principal Component Analysis (PCA) and Fisher Linear Discriminant Analysis (LDA). Quantitative measures of similarity are obtained and then used as inputs to several classification methods. In the classifying step, we use: Neural Networks (NN), k-Nearest Neighbor (KNN) and Support Vector Machines (SVM). To test this method and evaluate its performance, a simulation series of experiments were performed on 3D Shape REtrieval Contest 2008 database (SHREC2008).

Keywords: 3D face recognition, geodesic distance, reference point, Principal Components Analysis, Linear Discriminant Analysis, fast marching, eikonal equation


1. Introduction

The task of recognizing human face with the help of a machine has been has attracted more attention in recent years. Biometric face recognition technology has received significant attention in the past several years due to its potential in different applications. Automated human face recognition was applied in different fields including automated secured access to machines and buildings, automatic surveillance, forensic analysis, fast retrieval of records from databases in police departments, automatic identification of patients in hospitals, checking for fraud or identity theft, and human-computer interaction [27].

In a face recognition system, the individual is subject to a varied contrast and brightness lighting background. This three-dimensional shape when it is part of a two-dimensional surface as is the case of an image can lead to significant variations [3]. The human face is an object of three-dimensional nature. This object may be subject of various rotations, not only flat but also space and also subject to deformations due to facial expressions. The shape and characteristics of this object also change over time [13].

Automatic face recognition based on the 2D images processing is well developed this last years, and several techniques have been proposed [4]. There are a methods of 3D face recognition based on the use of three-dimensional information of the human face in the 3D space. Existing approaches that address the problem of 3D face recognition can be classified into several categories of approaches: Geometric or Local approaches 3D, Bronstein et al [1,2] propose a new representation based on the isometric nature of the facial surface, Samir et al [3,4] use 2D and 3D facial curves for analyzing the facial surface; Holistic approaches, Heseltine et al [5] have developed two approaches to applying the representations PCA Three-dimensional face, Cook et al [6] present a robust method for facial expressions based on Log-Gabor models from images of deep and some approaches are based on face segmentation can be found in [7-12].

The objective of this paper is to perform an automatic 3D face recognition system based on geodesic distance computing using Eikonal equation. For this we take the following steps:

- Detection of 3D face where the nose end is a reference point.
- Compute of geodetic distance between the reference point and the other points of the 3D facial surface using the Fast Marching algorithm as a solution of the Eikonal equation.
- Reduction of geodesic distances space matrices by Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms.

The rest of this paper is organized as follows: Section 2 describes the methodology of the proposed method with its stages: reference point detection, geodesic distance computing, Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms. Section 3 includes the simulation results and method analysis. Section 4 draws the conclusion of this work and possible points for future work.
2. Methodology

Given an image of 3D face Shape Retrieval Contest 2008 database (SHRED2008) our goal is to realize an automatic 3D face recognition system based on the computation of the geodesic distance between the reference point (nose tip) and the other points in the 3D face surface. So, our algorithm is divided to four main steps, first: Reference Point Detection, In this work we have detected the point of reference (nose tip) manually. Second: Geodesic Distance computation, an effective method to compute a geodesic distance between two points of facial surface is using the Fast Marching as a numerical algorithm for solving the Eikonal equation. Third: Dimensionality Reduction, in this step we use two known algorithms: Principal Component Analysis (PCA) and Fisher Linear Discriminant Analysis (LDA). Finally: classification algorithms, in this step we use three types of classification algorithms: the Neural Networks (NN), k-Nearest Neighbor (KNN) and Support Vector Machines (SVM). Figure (1) illustrates the steps of proposed method in our 3D face recognition system.

### 2.1. Reference Point Detection

The reference point (nose tip) is detected manually or automatically. There are several automatic approaches: L. Ballihi et al are developed an automatic algorithm nose end detection of a 3D face in [13]. This algorithm is based on two cuts of facial surface. The first is the transverse face of the mass center. The second cut is based on the minimum depth point of the horizontal curve obtained by the first cut. The output of the last cut is a vertical curve and the minimum depth of this curve is the end of 3D face nose. In [14] S. Bahamin et al use Gabor filters to automatically detect the nose tip. Another method has been used by C. Xu et al in [15], this method computes the effective energy of each neighbor pixel, then be determined the mean and variance of each neighbor pixel and uses the SVM to specify point end of the nose. L.H. Anuar et al [16] use a local geometry curvature and point signature to detect a nose tip region in 3D face model. In this work we have detected the point of reference \( p_0 \) (nose tip) manually. The following figure (Figure 2) summarizes the steps to detect the nose tip of a 3D face of an image of the database SHREC2008.

### 2.2. Geodesic Distance

The geodesic distance between two points' \( p_0 \) and \( p \) of 3D face surface is the shortest path between the two points while remaining on the facial surface. In the context of calculating the geodesic distance R. Kimmel and J.A. Sethian [17] propose the method of Fast Marching as a solution of the Eikonal equation.

Eikonal equation is of the form:

\[
|\nabla_{\Omega} (x)| = F(x); x \in \Omega
\]

With:
- \( \Omega \) is an open set in \( \mathbb{R}^n \) housebroken limit.
- \( \nabla \) denote the gradient.
- \( || \) is the Euclidean norm.

The Fast Marching method is a numerical method for solving boundary value problems of the Eikonal equation [17,18,19]. The algorithm is similar to the Dijkstra's algorithm [20] and uses that information flows only to the outside from the planting area.

### Figure 1. Methodology Architecture.

### Figure 2. 3D face nose end detection steps: (a) 3D face image; (b) Manual nose tip selection; (c) Reference point detection

### Figure 3. 3D facial surface discretized on triangular mesh of N vertices
We consider a 3D face surface discretized using a triangular mesh with N vertices. Figure 3 shows a 3D face image of the Shape REtrieval Contest 2008 (SHREC2008) database whose obverse surface is discretized into a triangular mesh.

The geodesic distance between two points on a surface is calculated as the length of the shortest path connecting the two points. Using the Fast Marching algorithm on the triangulated surface 3D face, we can compute the geodesic distance between the reference point \( P_0 \) and the other point’s \( p \) constructing the facial surface.

The geodesic distance \( \delta_{p_0, p} \) between \( p_0 \) and \( p \) is approximated by the following expression:

\[
\delta_{p_0, p} = \min_{\gamma} \gamma(\beta(p_0, p))
\]  

(2)

With:
- \( \beta(p_0, p) \) is the path between \( p_0 \) and according to the surface \( S \) of the 3D face.
- \( \gamma(\beta(p_0, p)) \) is the path length.

The following figure (Figure 4) shows the steps for determining the geodesic distance using a 3D face image of SHREC2008 database.

Repeating this computation (geodesic distance \( \delta_{p_0, p} \)) between the reference point \( p_0 \) and each point \( p \) of the surface \( S \) of the 3D face, then we compute a geodesic distance matrix \( \Psi \):

\[
\Psi = \begin{bmatrix}
\delta_{11} & \cdots & \delta_{1m} \\
\vdots & \ddots & \vdots \\
\delta_{n1} & \cdots & \delta_{nm}
\end{bmatrix}
\]

(3)

With, \( \delta_{ij} = \delta_{p_0, p} \).

To realize our 3D face recognition system, we use the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for writing space of the geodesic distance matrix \( \Psi \).

### 2.3. Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is to express a set of variables into a set of linear combinations of uncorrelated factors together, these factors account for a fraction of increasingly lower variability of the data. This method is used to represent the original data in a space of dimension less than the original space while minimizing the loss of information.

The recognition is performed by comparing the projection coefficients of a test image with those in the electronic driving components. The performance of the PCA will be illustrated by the following stages:

- We load the data to be compared in a matrix \( \Psi \).
- We thus determine the size of the data set:
  \[
  [n \times m] = \text{size} [\Psi]
  \]

(3)

- To summarize the data, we compute the sample mean vector and the sample standard deviation vector:

\[
\Psi \text{ mean} = \text{mean} (\Psi) \quad \text{and} \quad \Psi \text{ Std} = \text{Std} (\Psi)
\]

(4)

- Normalize the data. Here, the calibration means subtracting the average sample of each observation, and dividing by their standard deviation. This center and measure the data. Sometimes there are good reasons to change or do not perform this step, but we recommend you normalizes unless you have a good reason. It is easy to run this step as follows:

\[
\Psi = (\Psi - \text{repmat} (\text{Moyen}, [n \times 1])) / \text{repmat} (\text{Std}, [n \times 1])
\]

(5)

- The data matrix \( \Psi \) is multiplied by its transpose to obtain a covariance matrix \( L \) as shown in equation (5):

\[
L = \Psi^T \Psi
\]

(6)

- In this step the values is computed using Matlab languages and the corresponding eigenvectors in the covariance matrix by the equation (6):

\[
[V \ D] = \text{eig}(L)
\]

(7)

Where \( V \) is an orthogonal matrix of specific vectors and \( D \) is a diagonal matrix of eigenvalues.

We classify the eigenvectors \( v_i \in V \) according to decreasing values \( d_i \in D \).

The eigenvectors matrix \( V \) represents the projection eigenspace.

- In this step we project the images vectors centered in eigenspace. For this, we must compute the scalar product of these pictures vectors along with the eigenvector matrix by:

\[
\text{Eigen Comp} = \Psi^T \Psi
\]

(8)
2.4. Equations Fisher Linear Discriminant Analysis (LDA)

The Linear Discriminant Analysis (LDA) of Fisher method proceeds according to the following steps:

2.4.1. Computing of within-class Scatter Matrix $S_w$

The within-class scatters matrix measure the amount of dispersion between the images in the same class. For the i-th class, the dispersion matrix $S_i$ is calculated as the sum of the covariance matrices of the images centered in this category:

$$S_i = \sum_{j=1}^{M_i} (x_j - \mu_j)(x_j - \mu_j)^T$$  \hspace{1cm} (9)

Where $\mu_j$ is the mean of images in the class and $x_j$ is vector image. The matrix of the within-class dispersion $S_w$ is the sum of all the dispersion matrices.

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{M_i} (x_j - \mu_j)(x_j - \mu_j)^T$$  \hspace{1cm} (10)

Where $c$ is the class’s number.

2.4.2. Computing of between-class Scatter Matrix $S_b$

As with tables and equations, figures should be set in one column if possible unless two-column display is essential. The resolution of graphics and image should be adequate to reveal the important detail in the figure.

$S_b$ between-class scatter matrix measures the amount of dispersion between classes. It computes the sum of the difference between the total average and the average of each class.

$$S_b = \sum_{i=1}^{c} \sum_{j=1}^{M_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

with $\mu_i = \left(\frac{1}{c}\right)\sum_{i=1}^{c} u_i$

Where $\mu_i$ is the mean of images in class $i$ and $\mu$ is the mean of all images.

2.4.3. Solve the Generalized Eigenvalue Problem

We calculate the eigenvalues and corresponding eigenvectors for the two dispersion matrices within-class and between-class by the following equation:

$$S_b^*U_k = \lambda_k S_w^*U_k$$  \hspace{1cm} (12)

Where $U_k$ is a matrix of eigenvectors and $\lambda_k$ is a matrix of eigenvalues.

The eigenvectors (each vector has the size of 1x100) are ordered in descending order of their eigenvalues. Finally we keep only the first C-1 eigenvectors. These C-1 eigenvectors form the Fischer projection base.

2.5. Classification

Let us consider the 3D human face images. First, we compute the geodesic distances matrix $[\Psi] = \delta_{ij}$ as described in subsection (2.2). Then, we apply the reduce space algorithms (ACP or ALD) to geodesic distances matrix $[\Psi]$, we obtain a low size matrix $[A] = A_{pq}$. In the end, we determine the input vectors $v_i$ of classification algorithms using the linearization of the matrix $[A]$. Figure 5 illustrates the steps of extracting the input vectors of the classifier algorithms used in our face recognition system.

![Figure 5](image)

3. Simulation Results

In this section we make a series of simulation to evaluate the effectiveness of the proposed approach. These results were performed based on SHREC 2008 database. This database contain 427 scans of 61 subjects, for each of these 61 subjects 7 different scans namely: two “frontal”, one “look-up”, one “look-down”, one “smile”, one “laugh” and one “random expression” [21,22].

The following table summarizes the results obtained from simulation:

<table>
<thead>
<tr>
<th>Classification Algorithms</th>
<th>NN</th>
<th>KPPV</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geodesic Distance + PCA</td>
<td>92.00%</td>
<td>90.70%</td>
<td>95.30%</td>
</tr>
<tr>
<td>Recognition Rate</td>
<td>06.00%</td>
<td>09.30%</td>
<td>02.70%</td>
</tr>
<tr>
<td>Error Rate</td>
<td>90.70%</td>
<td>88.50%</td>
<td>93.20%</td>
</tr>
<tr>
<td>Geodesic Distance + LDA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recognition Rate</td>
<td>09.30%</td>
<td>11.50%</td>
<td>05.80%</td>
</tr>
<tr>
<td>Error Rate</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1 summarizes the simulation results obtained by the presented approach. This results shows the recognition rate and the error rate found by methods used in our 3D face recognition system: (Geodesic Distance + PCA) and (Geodesic Distance + LDA) for characters extraction, with three algorithms used in classification such as: Neural Networks (NN), k-Nearest Neighbor (KNN) and Support Vector Machines (SVM).
The results obtained by our 3D face recognition system shows that our method present the best recognition rate (95.30%) using (Geodesic Distance + PCA) for images of SHREC 2008 database.

In conclusion of this series of results, a summary table (Table 2) compares the performance of our face authentication with respect to the performance obtained in other 3D face recognition systems.

<table>
<thead>
<tr>
<th>Date</th>
<th>Reference</th>
<th>Method</th>
<th>Database</th>
<th>Reported performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2004</td>
<td>Haar et al [26]</td>
<td>facial Contour Curves</td>
<td>SHREC’08</td>
<td>91.1 %</td>
</tr>
<tr>
<td>2007</td>
<td>Feng et al [24]</td>
<td>Euclidean Integral Invariants Signature (Closed 3D Curves)</td>
<td>FRGCv2</td>
<td>95.0 %</td>
</tr>
<tr>
<td>2007</td>
<td>Samir et al [23]</td>
<td>Planar Curves Levels</td>
<td>Notre Dame</td>
<td>90.4 %</td>
</tr>
<tr>
<td>2008</td>
<td>Daoudi et al [25]</td>
<td>Elastic Deformation Of Facial Surfaces (open paths)</td>
<td>FSU</td>
<td>92.0 %</td>
</tr>
<tr>
<td>2015</td>
<td>Our System</td>
<td>GD+LDA+SVM</td>
<td>SHREC’08</td>
<td>95.3 %</td>
</tr>
</tbody>
</table>

We can notice that the performance of our automatic 3D face recognition system, In addition our system is perfect in all assessment. Our goal was to improve 3D faces recognition system we affirm based on the results that our goal is achieved.

4. Conclusion

In this paper, we have presented a 3D face recognition system based on the computation of the geodesic distance between the reference point and the other points in the 3D face surface. For writing space we have used PCA and LDA. For the classifying step we have implemented algorithms as NN, KNN and SVM. Simulation results give us a better recognition rate (95.3%) for the (ACP + DG + SVM) method. In future work, we will take a 3D face recognition system by analyzing of iso-geodesic curves using Riemannian geometry and compare these results with those obtained by this method.

References


