International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:9, No:6, 2015

Methods of Geodesic Distance in Two-Dimensional Face Recognition

Rachid Ahdid, Said Safi, Bouzid Manaut

Abstract**—**In this paper, we present a comparative study of three methods of 2D face recognition system such as: Iso-Geodesic Curves (IGC), Geodesic Distance (GD) and Geodesic-Intensity Histogram (GIH). These approaches are based on computing of geodesic distance between points of facial surface and between facial curves. In this study we represented the image at gray level as a 2D surface in a 3D space, with the third coordinate proportional to the intensity values of pixels. In the classifying step, we use: Neural Networks (NN), K-Nearest Neighbor (KNN) and Support Vector Machines (SVM). The images used in our experiments are from two well-known databases of face images ORL and YaleB. ORL data base was used to evaluate the performance of methods under conditions where the pose and sample size are varied, and the database YaleB was used to examine the performance of the systems when the facial expressions and lighting are varied.

Keywords**—**2D face recognition, Geodesic distance, Iso-Geodesic Curves, Geodesic-Intensity Histogram, facial surface, Neural Networks, K-Nearest Neighbor, Support Vector Machines.

I. Introduction

FACE recognition has been a very hot research topic in recent years, automatic face recognition has become a popular area of research and several methods of 2D and 3D face recognition are developed. Automatic verification and identification of faces from still images or video data can be seen as a pattern recognition problem, which is very hard to solve due to its nonlinearity. 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 [1]. 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 [2].

Automatic human faces recognition based on the 2D images processing is well developed this last years, and several techniques have been proposed. Many methods have been proposed for face recognition within the last years. Among these methods: In 1991 M. A. Turk et al. implemented The Principal Component Analysis (PCA) approach also known under the name Eigenfaces [3]. In 1999 A. Nefian described an embedded Hidden Markov Model (HMM) based approach for face detection and recognition that uses an efficient set of observation vectors obtained from the 2D-DCT coefficients [20]. Two-dimensional version of Principal Component Analysis noted (2DPCA) was presented by J. Yang et al. in 2004 for image representation [4]. In 2007, P. Yan et al. and H. Chen et al. used Linear Discriminant Analysis LDA also known under the name Fisherfaces [5], [6]. M. Visani et al. are proposed Two-Dimensional Linear Discriminant Analysis (2DO-LDA) in 2004, this approach is chosen to jointly maximize the mean variation between classes and minimize the mean of the variations inside each class [7]. In 2012 M. Belahcene et al. present a face recognition hybrid method constructed by (PCA + EFM + 200SVMs), constructed by: Principal Component Analysis PCA, Discriminant Linear Model Improved Fisher and 200 Support Vector Machines SVM for classification [9]. H. Cevikalp et a.l proposed an approach called the Discriminative Common Vector method based on a variation of Fisher's Linear Discriminant Analysis for the small sample size case in 2005 [8]. In 2003 J. Lu et al. implemented a method combines the strengths of the D-LDA and F-LDA approaches, while at the same time overcomes their shortcomings and limitations [10]. M.S. Bartlett et al. and B.A. Draper et al. proposed using ICA for face representation and found that it was better than PCA, respectively in 2002 and 2003 [11], [12]. In 2014 W. Xu et al. proposed an integrated algorithm based on the respective advantages of wavelets transform (WT), 2D Principle Component Analysis (PCA) and Support Vector Machines (SVM) [13].

The paper presents face recognition approaches using Riemannian geometry. The objective of this works is to compare three methods of two-dimensional face recognition system based on computing of geodesic distance. For this, we take the following steps:

- Detection of 2D face where the nose end is a reference point.
- Compute of geodetic distance between the reference point and the other points of the 2D 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.
- Determination of the Geodesic-Intensity Histogram (GIH). It captures the joint distribution of the geodesic distance and the intensity of the sample points.
- Iso-Geodesic Curves extraction using a Fast Marching algorithm. Compute of geodetic distance between two iso-

Rachid Ahdid is with the Department of Mathematics and Informatics, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco. (Corresponding author: e-mail: r.ahdid@ usms.ma).

Said Safi is with the Department of Mathematics and Informatics, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco.

Bouzid Manaut is with the Departement of Physics, Polydisciplinary Faculty, Sultan Moulay Slimane University, Beni Mellal, Morocco.

geodesic curves using mathematical formulas in Riemannian metric.

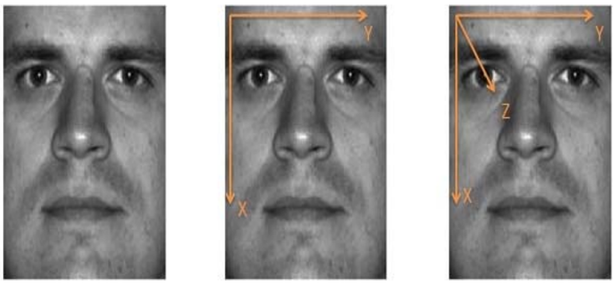
The remainder of the paper is organized as follows: in Section II we present the used methods in our 2D face recognition system: Geodesic Distance computing (GD), Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Geodesic-Intensity Histogram (GIH) and Iso-Geodesic Curves (I-GC). In Section III, we describe the data sets (ORL and YaleB) and experimental results. Finally, we formulate our conclusions in Section IV.

II. Proposed Methods

A. Geodesic Distance

1) Preprocessing

A face image preprocessing step is first applied to each image: it consists in centering the face in the image, normalize for assets of homogeneous sizes, equalizing its histogram and treat also the intensity images as 2D surfaces in 3D space with the third coordinate proportional to the intensity values of pixels. Fig. 1 illustrates an example of a detected image preprocessing.

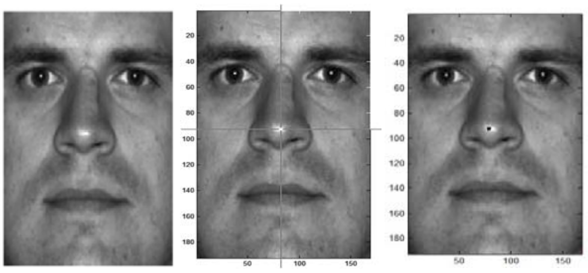


(a) (b) (c)

Fig. 1 Preprocessing of an image of YaleB database: (a) Face image detected; (b) grayscale and normalized 2D images; (c) 2D face surfaces in 3D space

2) Reference Point Detection

The reference point (nose tip) is detected manually or automatically. In this work we have detected the reference point p_0 (nose tip) manually. Fig. 2 summarizes the steps to detect the nose tip of a 2D face image.



(a) (b) (c)

Fig. 2 Reference point detection steps: (a) 2D face image after preprocessing; (b) Manual nose tip selection; (c) Reference point detection

3) Geodesic Distance

The geodesic distance between two points' p_0 and p of 2D 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 [21] propose the method of Fast Marching as a solution of the Eikonal equation.

The Eikonal equation is of the form:

$$|\nabla_u(x)| = F(x);\ x \in \Omega \qquad (1)$$

with: Ω is an open set in R^n housebroken limit. ∇ denotes the gradient. |.| is the Euclidean norm.

The Fast Marching method is a numerical method for solving boundary value problems of the Eikonal equation [21]-[23]. The algorithm is similar to the Dijkstra's algorithm [24]. In this work, we compute a geodesic distance on a facial surface, using the values of the surface gradient only [25].

The main step of the geodesic distance computing is the construction of the canonical form of a given surface (Facial surface). Let Img a 2D face image of ORL or YaleB database, we can represent mathematically Img as a plan P = (x,y). To compute a geodesic distance the facial surface can be thought of as a parametric manifold, represented by a mapping F: R2 → R3 from the parameterization plan P(x,y) to the manifold [25]:

$$F(P) = F(x , y) = (x , y , z_{(x,y)}) \qquad (2)$$

The metric tensor g_{ij} of the manifold is given by:

$$g_{ij} = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} X.X & X.Y \\ Y.X & Y.Y \end{bmatrix} \qquad (3)$$

with the inner product defined as follows:

$$X.X = \|X\|.\|X\|.\cos(0) \qquad (4)$$

$$X.Y = \|X\|.\|Y\|.\cos(\alpha) \qquad (5)$$

$$Y.X = \|Y\|.\|X\|.\cos(\beta) \qquad (6)$$

$$Y.Y = \|Y\|.\|Y\|.\cos(0) \qquad (7)$$

and,

$$\cos(\alpha) = \frac{X.Y}{\|X\|.\|Y\|} = \frac{g_{12}}{\sqrt{g_{11}.g_{22}}} \qquad (8)$$

$$\cos(\beta) = \frac{Y.X}{\|Y\|.\|X\|} = \frac{g_{21}}{\sqrt{g_{22}.g_{11}}} \qquad (9)$$

The angle between the non-orthogonal axes is calculated by:

$$\alpha = \cos^{-1}\left[\frac{g_{12}}{\sqrt{g_{11}.g_{22}}}\right] = Arccos\left[\frac{g_{12}}{\sqrt{g_{11}.g_{22}}}\right] \qquad (10)$$

$$\beta = \cos^{-1}\left[\frac{g_{21}}{\sqrt{g_{22}.g_{11}}}\right] = Arccos\left[\frac{g_{21}}{\sqrt{g_{22}.g_{11}}}\right] \qquad (11)$$

If $\alpha = \beta = \frac{\pi}{2}$, the axes are perpendiculars.

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 surface gradient, 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 $\delta_{p0,p}$ between two points' p_0 and p is approximated by:

$$\delta_{p0,p} = \min \gamma(\beta(p_0, p)) \qquad (12)$$

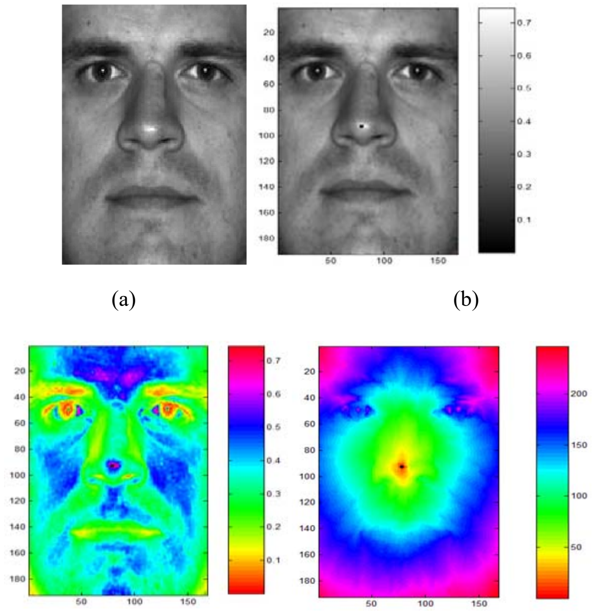
with: $\beta(p_0, p_i)$ is the path between p_0 and according to the surface S of the 3D face. $\gamma(\beta(p_0, p))$ is the path length.

The distance element on the manifold is given by [26]:

$$\delta_{ij} = \sqrt{g_{ij}\,\xi^i\,\xi^j} \qquad (13)$$

with: g_{ij} is computed by (3); i= 1 or 2 and j = 1 or 2; $\xi^1 = x$ and $\xi^2 = y$.

Fig. 3 shows the steps for determining the geodesic distance using a 3D face image of YaleB database.



(a) (b)

(c) (d)

Fig. 3 2D face geodesic distance computing Steps: (a) 2D face image; (b) Reference point detection; (c) Color-based representation of 2D surface in 3D space; (d) Geodesic distance computing.

Repeating this computation (geodesic distance $\delta_{p0,p}$) between the reference point p_0 and each point p of the surface S of the 3D face, then we obtain a high-dimensional matrix Ψ of geodesic distances:

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

Since higher the dimension of the space is, more the computation we need to find a match, a dimensional reduction technique is used to project the problem in a lower-dimensional space. In this work, we use the Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) algorithms as dimensionality reduction techniques.

B. 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 [3], [4].

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:

- We load the data to be compared in a matrix Ψ.
- We thus determine the size of the data set:

$$[n\ m] = \text{size}\ [\Psi] \qquad (14)$$

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

$$\Psi\ \text{mean} = \text{mean}\ (\Psi)\ \text{et}\ \Psi\text{Std} = \text{Std}\ (\Psi) \qquad (15)$$

- 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:

$$\Psi = (\Psi\ \text{-repmat}\ (\Psi\ \text{Moyen},[n\ 1]))./\text{repmat}(\Psi\ \text{Std},\ [n\ 1]) \qquad (16)$$

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

$$L = \Psi * \Psi^T \qquad (17)$$

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

$$[V\ D] = eig(L) \qquad (18)$$

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:

$$EigenComp = \Psi * V \tag{19}$$

C. Fisher Linear Discriminant Analysis (LDA)

The Linear Discriminant Analysis (LDA) of Fisher method proceeds according to the following steps [5]-[7], [19]:

1) Computing of Within-Class Scatter Matrix S_w

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

$$S_i = \sum_{j=1}^{Mi}(x_j - u_j)\ (x_j - u_j)^T \tag{20}$$

where μ_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.

$$S_w = \sum_{i=1}^{c}\sum_{j=1}^{Mi}(x_j - u_j)\ (x_j - u_j)^T \tag{21}$$

where c is the class's number,

2) Computing of Between-Class Scatter Matrix S_b

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.

$$S_b = \sum_{i=1}^{c}(u_i - u)(u_i - u)^T \text{ With } u = (\tfrac{1}{c})\sum_{i=1}^{c} u_i \tag{22}$$

where μ_i is the mean of images in class i and μ is the mean of all images.

3) Solve the Generalized Eigenvalue Problem

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

$$S_b . U_k = \lambda_k . S_W . U_k \tag{23}$$

where, U_k is a matrix of eigenvectors and λ_k is a matrix of eigenvalues.

D. Geodesic Intensity Histogram

Geodesic Intensity Histogram is a deformation invariant descriptor extracted from the echantionage geodesic, it captures the joint distribution of the geodesic distance and intensity of points.

Let a 2D face image of ORL or YaleB database, after the preprocessing step and the detection of a reference point p_0. we determine the GIH matrix using a geodesic distance computing (between p_0 and other points p of the 2D face surface) and the pixel intensity of points p.

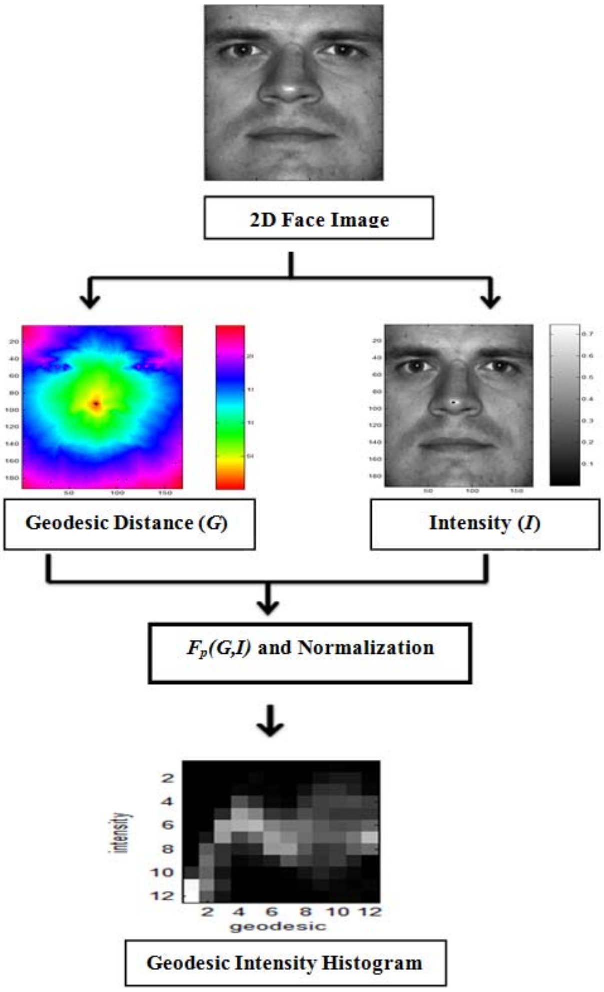


Fig. 4 Diagram of Geodesic Intensity Histogram steps

GIH is a two-dimensional normalized histogram obtained by the following steps [28]:

- Computation of geodesic distance *G(p)* and determination of pixel intensity *I(p)*.
- Divide the 2D face image (intensity and geodesic distance) space into *N×M*. *N* is the number of intensity intervals, and *M* the number of geodesic distance intervals.
- Determination of $F_p(G, I)$ using (24):

$$\forall 1 \leq k \leq K \text{ and } \forall 1 \leq m \leq M$$

$$F_p(G, I) = \{p \in F_p : (I(p), g(p)) \in B(n, m)\} \tag{24}$$

with, B(n, m) is the bin corresponding to the n_{th} intensity interval and the m_{th} geodesic interval; G(p) and I(p) are respectively the geodesic distance and the intensity at p.

- Normalization of $F_p(G, I)$.

E. Facial Curves

This 2D face recognition method is based on the analysis of facial surfaces by analyzing of facial curves using Riemannian geometry. To extract the curves of a 3D face surface, the first step is to define the real-valued function on this surface [27], [29], [30]. Iso-geodesic curves are defined as the locations of all points on the facial surface having the same geodesic distance to the reference point chosen (end of nose). The geodesic distance between two points on a surface is the shortest path between these two points along of surface [30], [31].

In this work, we represent the 2D human face surface by a collection of iso-geodesic curves. To extract the iso-geodesic curves we use the Fast Marching algorithm as a solution of Eikonal equation [21]. Fig. 4 presents the steps of extracting of iso-geodesic curves in some 2D face images of YaleB and ORL databases.

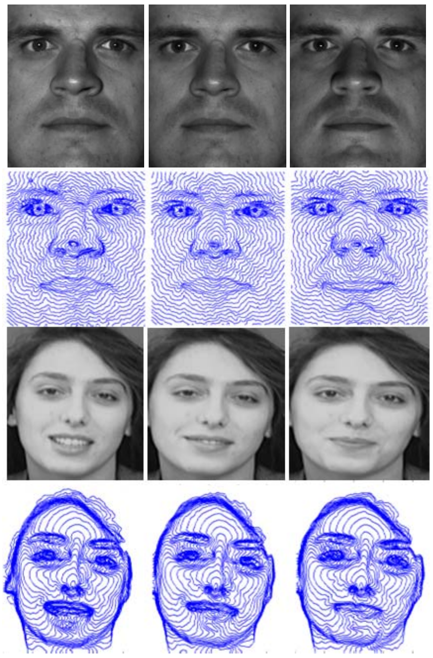


Fig. 5 Iso-Geodesic Curves extracting using ORL and YaleB images

Given two points on a face surface S (reference point p_0 and p), A geodesic distance between p_0 and p is defined as the arc length of the shortest path between these two points along the surface and denoted by a Geodesic Distance Function (GDF), which is a continuous function on the facial surface.

$$F\ (p,\ p_0) = k\ ;\ \ k \in [0,+\infty[\ \text{ and } (p_0,p) \in S \qquad (25)$$

We can therefore define the facial curves by:

$$C_k = \{p \in S \setminus F(p,\ p_0) = k\} \subset S\ ,\ k \in [0,+\infty[\qquad (26)$$

The function F defines the geodesic distance between p_0 and p, or the length of the shortest path between these two points while remaining on the surface S.

This definition allows us to cite three cases c_k according to the values of k:

- If k = 0 then c_k tends towards the reference point p_0 : $c_k = \{ p_0 \}$.
- If $k \rightarrow \infty$ then c_k is empty: $c_k = \{ \emptyset \}$.
- If $0 < k < \infty$ then c_k approaches S: $c_k = \{p \in S \setminus F(p,p_0) = k \}$.

To analyze the facial surfaces, we simply analyze the iso-geodesic curves that characterize these 3D face surfaces and compute a geodesic distance between them on a manifold depends on the Riemannian metric. To analyze the curve shape, we use the parameterization by the mathematical function SRVF (Square Root Velocity Function) [32]-[36].

Let a parameterized closed curve be denoted as $\beta: I \rightarrow \mathbb{R}^3$, for a unit interval $I \equiv [0, 2\pi]$, β is represented by its SRVF: $q: I \rightarrow \mathbb{R}^3$ defined as:

$$q(s) = \frac{\dot{\beta(t)}}{[\|\dot{\beta}(s)\|]^{1/2}} = \frac{\frac{d\beta(s)}{ds}}{\sqrt{\|\frac{d\beta(s)}{ds}\|}} \in \mathbb{R}^3 \qquad (27)$$

where, $s \in I \equiv [0, 2\pi]$. $\|.\|$ is the standard Euclidean norm in $\mathbb{R}^3$. $\|q(s)\|$ is the square-root of the instantaneous speed on the curve β. $\frac{q(s)}{\|q(s)\|}$ is the instantaneous direction at the point $s \in [0,\ 2\ \pi]$ along the curve. Thus, the curve β can be recovered within a translation, using:

$$\beta(s) = \int_0^s q(t)\|q(t)\|dt \qquad (28)$$

We define the set of closed curves {β} in $\mathbb{R}^3$ by:

$$C = \left\{q: S^1 \rightarrow R^3 \mid \int_{S^1} q(t)\|q(t)\|dt = 0\right\} \subset L(S^1, R^3) \qquad (29)$$

where, $L^2(S^1, \mathbb{R}^3)$: denotes the set of all functions integral S^1 to $\mathbb{R}^3$. $\int_{S^1} q(t)\|q(t)\|dt$: denotes the total displacement in $\mathbb{R}^3$ while moving from the origin of the curve until the end. When $\int_{S1} q(t)\|q(t)\|dt = 0$, the curve is closed.

All 3D closed curves are defined as nonlinear variety in the Helbert space. To analyze the shapes of the iso-geodesic curves and compute a geodesic distances between them, it is important to understand all vectors of their tangent spaces and impose a Riemannian structure. We equip the space of the closed curves of a Riemannian structure using the inner product defined as [33], [34], [36]:

$$< f, g > = \int_0^1 (\ f(s), g(s)\ )ds \qquad (30)$$

Here, f and g are two vectors in the tangent space $T_v(c)$. We can also define $T_v(c)$:

$$Tv(C) = \left\{f: S^1 \rightarrow R^3 | < f(s), h(s) >= 0,\ \ h \in N_v(c)\ \right\} \qquad (31)$$

$N_v(c)$ is a space of the normal vectors to the face curve.

After a mathematical representation of iso-geodesic curves using Riemannian metric, this metric should invariant certain transformation (translation, rotation, scale) [34]. The question to ask is how to compute the geodesic distance between two closed curves. To answer this question, we used the approach introduced by Klassen et al. in 2007 [33], [34]. This method use path straightening flows to find a geodesic between two shapes.

To compare two facial surfaces, we just compare a pairs of closed curves of these two facial surfaces. Lets c_1 and c_2 two facial curves (iso-geodesic curves), q_1 and q_2 are respectively there Square Root Velocity Function (SRVF). The geodesic distance between c_1 and c_2 is computed by:

$$d(q_1, q_2) = \int_0^1 \sqrt{< \dot{\varepsilon}(t), \dot{\varepsilon}(t) >}\,\mathrm{dt} \tag{32}$$

with $\boldsymbol{\varepsilon}$ is a geodesic path determined by the training method, this method is to connect the two curves by an arbitrary path α then update the path repeatedly in the negative direction of the gradient of the energy given by:

$$\mathrm{E}[\alpha] = \frac{1}{2}\int_0^1 < \frac{\mathrm{d}}{\mathrm{ds}}\alpha(\mathrm{t}), \frac{\mathrm{d}}{\mathrm{ds}}\alpha(\mathrm{t}) > \mathrm{dt} \tag{33}$$

$\boldsymbol{\varepsilon}$ has been shown that the critical points of the energy equation E(α) are geodesic paths in S [31], [32], [34].

The facial surfaces S_1 and S_2 are represented by their iso-geodesic curves collection respectively $\{c_k^1;\ \mathrm{k} \in [0,\ \mathrm{k}_0]\}$ and $\{c_k^2;\ \mathrm{k} \in [0,\ \mathrm{k}_0]\}$, k is a geodesic distance between p_0 (reference point) and p two points of facial surface S. The vectors of geodesic distances computed between a pairs of facial curves are used as input vectors of classification algorithms of our automatic facial recognition system.

III. 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 images of ORL and YaleB databases. These databases were used to evaluate the performance of this method with different simple size of laying and of the sample size are varied. The ORL database contains 400 images of 40 individuals. For each person, we have 10 pictures in grayscale and standardized at a resolution of 112×92 pixels. The YaleB database contains 2432 images of 38 people in 64 different lighting conditions. Each image has been normalized at a resolution of 168×192 pixels. To realize our 2D face recognition systems, we use three classification algorithms such as: the Neural Networks (NN), K-Nearest Neighbor (KNN) and Support Vector Machines (SVM).

A. Geodesic Distance (GD)

In the first experiment, we realize a 2D face recognition system based using three algorithms such as: Geodesic Distance (GD), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Given a face candidate of ORL (YaleB) database images, the geodesic distances between a reference point p_0 and other points of facial image p are computed as described in subsection (II-*A*). We obtain a geodesic distance matrix $[\Psi] = \delta_{ij}$ of high-dimensional 112×92 using ORL database images and 168×192 using YaleB database. The PCA and LDA algorithms are used for dimensionality reduction of $[\Psi]$ to find the vector which best account. These vectors define the input of the classification algorithms used in our 2D face recognition systems. Fig. 6 shows the experiment results of recognition rate obtained for YaleB and ORL images using a Geodesic Distance and Principal Component Analysis (GD+PCA) for feature extraction step. For classification step we are used tree algorithms (NN, KNN and SVM).

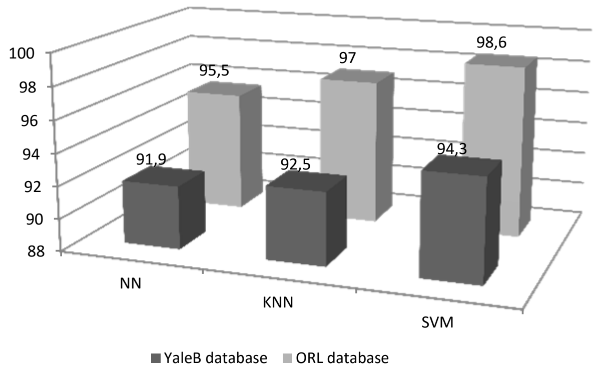


Fig. 6 Recognition Rate for YaleB and ORL images using GD+PCA and tree classification algorithms (NN, KNN and SVM)

The experiment results of this method indicate that the best recognition rate was obtained using Support Vector Machines (SVM) as classification algorithm with 98,60% for ORL images and 94,80% for YaleB images. In other experiment, the features were extracted using Geodesic Distance and Linear Discriminant Analysis (GD+LDA). The simulation results of this method are presented in Fig. 7.

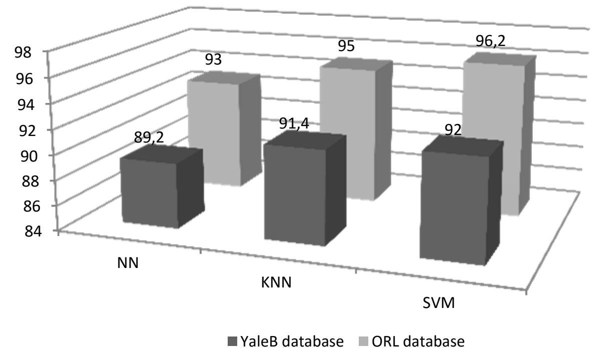


Fig. 7 Recognition Rate for YaleB and ORL images using GD+LDA and tree classification algorithms (NN, KNN and SVM)

The results obtained by our 2D face recognition system shows this method (GD+LDA) present the best recognition rate using SVM classification algorithm with 96,20% and 92,00%, respectively for images of ORL and YaleB databases.

B. Geodesic Intensity Histogram (GIH)

In the second experiment, we use a Geodesic Intensity Histogram (GIH) to features extraction of 2D face images. Let a 2D face image of ORL or YaleB database, the GIH is computed as described in subsection (II-*B*). The GIH vectors define the input of the classification algorithms used in our 2D face recognition systems. In classification steps we applied three weak classifiers, namely, the Neural Networks (NN), k-Nearest Neighbor (KNN) and Support Vector Machines (SVM). We achieved average recognition rates of 90,60%, 92,00% and 94,50%, respectively using ORL data and 90,00%, 91,00% and 93,70%, respectively for YaleB data. We summarize the resulting recognition rates in Fig. 8.

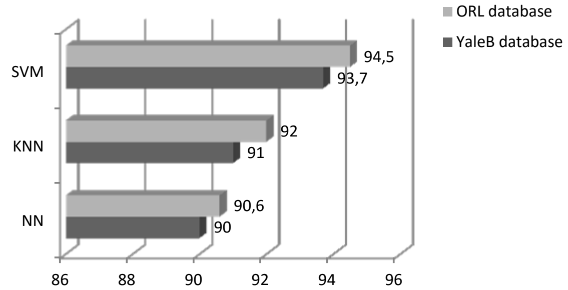


Fig. 8 Recognition Rate for YaleB and ORL images using GIH and tree classification algorithms (NN, KNN and SVM)

C. Iso-Geodesic Curves (I-GC)

In the last experiment, the features were extracted using Iso-Geodesic Curves (I-GC). This method was based on two principal steps: iso-geodesic curves extraction using Fast Marching algorithm as solution of Eiconal equation and compute the length of the geodesic path between each facial curve and its corresponding curve using a Riemannian framework.

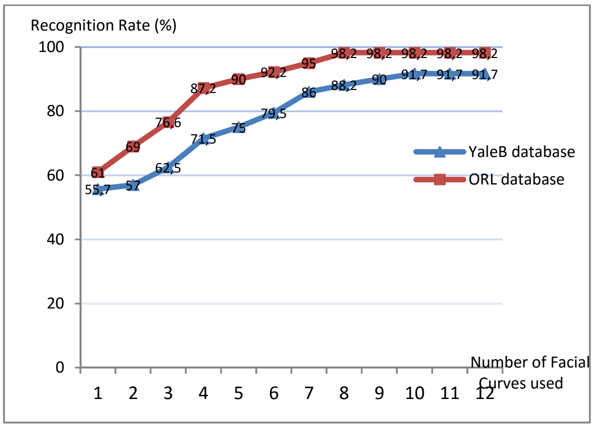


Fig. 9 Recognition rate in terms of number of facial curves used to represent the human face of ORL and YaleB database

Fig. 9 shows the recognition rate in terms of number of facial curves used to represent a 2D human faces used in our systems. This figure shows that the images of the ORL database are represented using eight facial curves and to represent images of the YaleB database, we must use eleven iso-geodesic curves.

Given a candidate 2D face image (Img) of ORL database (YaleB database). Img is represented using eight iso-geodesic curves (eleven iso-geodesic curves). The shortest path between two 2D face images id defined as the sum of the distance between all pairs of corresponding facial curves in the two face images. The feature vector is then formed by the geodesic distances computed on all the curves and its dimension is equal to the number of used iso-geodesic curves (8 for ORL data and 11 for YaleB data). These vectors are used as input of classification algorithms of our 2D face recognition system.

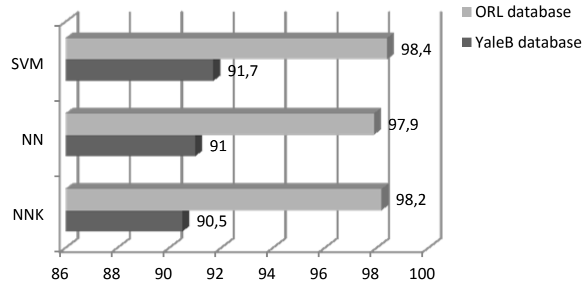


Fig. 10 Recognition Rate for YaleB and ORL images using I-GC and tree classification algorithms (NN, KNN and SVM)

Fig. 10 shows the recognition rate for images of ORL and YaleB databases. The best recognition rate was obtained using Support Vector Machines (SVM) as classification algorithm and equal 98,40% for ORL database and 91,70% for YaleB images.

D. Comparison of Experiment Results

In this paper, a number of face recognition algorithms have been described. These methods have been verified on the ORL and YaleB dataset, and the testing protocols used in the experiments are almost the same, so that a direct comparison of the results reported in these works is possible. In Figs. 11 and 12, we give a comparison of these face recognition algorithms.

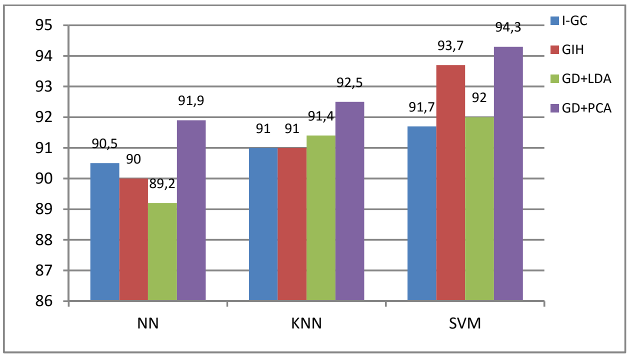


Fig. 11 Comparison of our 2D face recognition methods using the YaleB Database

Fig. 11 gives a comparison recognition rate of these four features extraction algorithms (GD+PCA, GD+LDA, GIH and I-GC) using YaleB database images. This comparison shows the best recognition rate (94.30%) was presented for GD+PCA using SVM classifier, then this method (GD+PCA) was also better than other tree approaches (GD+LDA, GIH and I-GC).

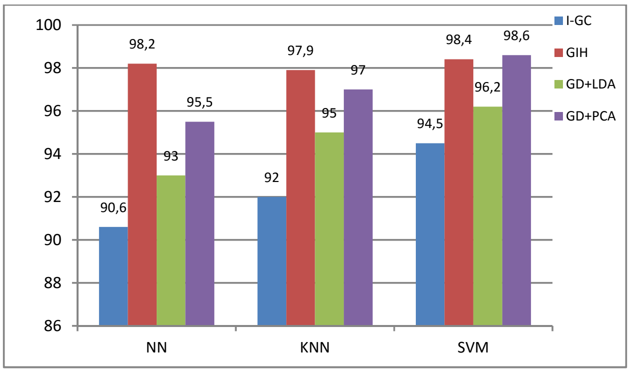


Fig. 12 Comparison of our 2D face recognition methods using the ORL Database

The comparison of features extraction algorithms using ORL images is given in Fig. 12. This comparison shows that the best approach is GD+PCA using SVM classifier with 98,60% in recognition rate.

The ORL database was used to evaluate the performance of our system algorithms (GD+PCA, GD+LDA, GIH and I-GC) under conditions where the pose, facial expressions and sample size are varied. The YaleB database was used to examine the system performance when both facial expressions and illumination are varied. The above experiments showed that the recognition rate of all algorithms using ORL images is always higher than YaleB images, then we can say the lighting conditions is the problem of this features extraction algorithms. Fig. 13 shows the problem that we find in our features extraction methods based on geodesic distance computing in using 2D face images of YaleB database.

Fig. 13 shows the limitation of our algorithms (GD, GIH and I-GC) using YaleB images: First line give three YaleB images with illumination are varied, second line shows the color Geodesic Distance computing, third line present the Iso-Geodesic Curves extraction and the last line give the Geodesic Intensity Histograms. These three methods were given the bad results when the intensity was lower.

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

We can notice that the performance of our automatic 2D face recognition system using Geodesic Distance (GD), Principal Component Analysis (PCA) and Support Vector Machines (SVM), In addition our system (GD+PCA+SVM) is perfect in all assessment. Our goal was to improve 2D faces recognition system we affirm based on the results that our goal is achieved.

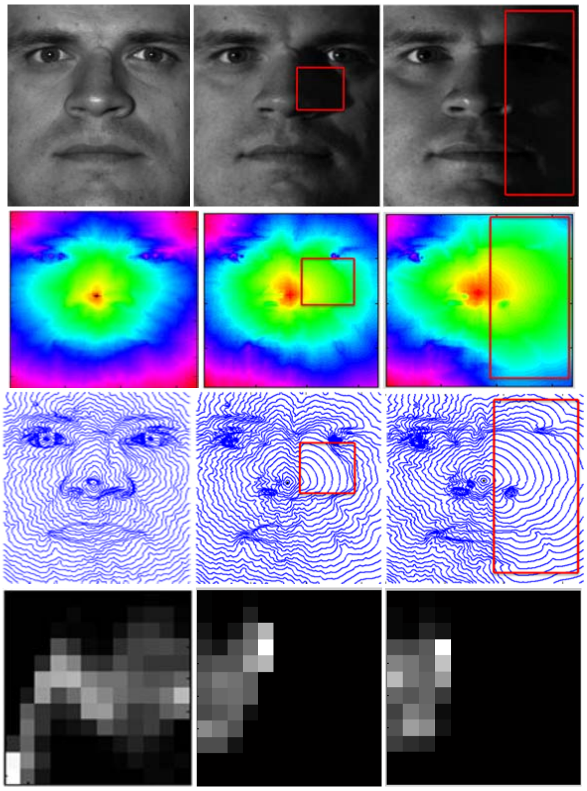


Fig. 13 The limitation of our algorithms (GD, GIH and I-GC) using YaleB images

IV. Conclusion

In this paper, we have presented an automatic 2D face recognition system using four feature extraction algorithms such as: GD+PCA, GD+LDA, GIH and I-GC. For classification step we used NN, KNN and SVM. The series of experiments were performed on two face image databases: ORL and Yale face databases. The recognition rate across all trials was higher using ORL images than YaleB images. The experimental results also indicated that the extraction of image features is computationally more efficient using GD+PCA algorithm than other approaches (GD+LDA, GIH and I-GC). The Support Vector Machine (SVM) classifier is better than Neural Networks (NN) and k-Nearest Neighbor (KNN) classifiers in terms of recognition accuracy in all experiments.

TABLE I
COMPARISON OF OUR METHODS WITH OTHER METHODS OBTAINED IN OTHER WORK SYSTEMS

Date	Reference	Method	Database	Reported performance
1991	M. Turk et al [3]	Eigenface	ORL	90,00%
2001	G.D. Guo et al [15]	Optimal-Pairwise Coupling (O-PWC) SVM	ORL	96,79%
2005	Cevikalp et al [8]	DCV	Yale	97.33%
2002	K. I. Kim et al [14]	Several SVM+NN arbitrator	ORL	97,90 %
2003	Lu et al[10]	DF-LDA	ORL	96.00%
2004	J. Yang et al [4]	2DPCA	ORL	96,00 %
2004	M. Visani et al [7]	2DO-LDA	FERET	94,40 %
2009	Salimi et al [16]	KPCA+LDA	XM2VTS	97,77 %
2010	M. Agarwal et al [17]	PCA+NN	ORL	97.01%
2012	M. Belahcene et al [9]	PCA+EFM+200 SVMs	XM2VTS	97,72 %
2012	V. More et al [18]	FFLD	ORL	95.50%
2012	V. More et al [18]	FFLD	Yale	94.80%
2014	W. Xu et al [13]	WT+ 2D PCA+ SVM	ORL	97.10%
2015	Our System	GD+LDA	YaleB	92.00%
2015	Our System	GD+LDA	ORL	96.20%
2015	Our System	GIH	YaleB	93.70%
2015	Our System	GIH	ORL	94.50%
2015	Our System	I-GC	YaleB	91.70%
2015	Our System	I-GC	ORL	98.40%
2015	Our System	GD+PCA	YaleB	94.80%
2015	Our System	**GD+PCA**	**ORL**	**98.60%**

REFERENCES

[1] C. Samir, A. Srivastava, M. Daoudi: "Three-dimensional face recognition using shapes of facial curves". IEEE Transactions on Pattern Analysis and Machine Intelligence, 1847–1857, (2006).

[2] L. Ballihi, B. Ben Amor, M. Daoudi, A. Srivastava, D. Aboutajdine: Sélection de courbes de la surface nasale pour l'authentification de personnes en utilisant Adaboost. hal-00666262, version 1 - 3 Feb 2012.

[3] M. Turk and A. Pentland, "Eigenfaces for Recognition," Journal of Cognitive Neuroscience, vol 3no 1, pp. 71-86. 1991.

[4] J. Yang., D. Zhang and D. Frangi: "Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition", IEEE Trans On PAMI, 26(1), pp: 131-137, 2004.

[5] P. Yan, K.W. Bowyer: "Biometric recognition using 3D ear shape". IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 8, pp. 1297–1308. (2007).

[6] H. Chen, B. Bhanu: Human ear recognition in 3D. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 4, pp. 718–737. (2007).

[7] M. Visani., C. Garcia and J.M. Jolion., "Two- Dimensional-Oriented Linear Discriminant Analysis for Face Recognition," Proc. of the Int. Conf. On Computer Vision and Graphics ICCVG'04, dans la srie Computational Imaging and Vision, Varsovie, Pologne, 2004.

[8] H. Cevikalp., M. Neamtu., M. Wilkes, A. Barkana.: Discriminative common vectors for face recognition. IEEE Trans. Pattern Anal. Machine Intell. 27 (1), 4–13. 2005.

[9] M. Belahcene., A. Ouamane., M. Boumehrez., and A. Benakcha., "Comparaison des mthodes de rduction d'espace et l'application des SVMS pour la classification dans l'authentification de visages," COURRIER DU SAVOIR, pp. 117-126, 2012.

[10] J. Lu, J., N. P. Kostantinos, N. V. Anastasios: "Face recognition using LDA-based algorithms". IEEE Trans. Neural Networks 14 (1), 195–200. 2003.

[11] B.A. Draper, K. Baek, M.S. Bartlett, J.R. Beveridge, "Recognizing Faces with PCA and ICA," Computer Vision and Image Understanding: special issue on face recognition, in press, pp. 115-137, 2003.

[12] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski, "Face Recognition by Independent Component Analysis," IEEE Trans. Neural Networks, vol. 13, no. 6, pp. 1450-1464, 2002.

[13] W. Xu and E. J. Lee, "Face Recognition Using Wavelets Transform and 2D PCA by SVM Classifier", International Journal of Multimedia and Ubiquitous Engineering, Vol.9, No.3, pp.281 -290, (2014).

[14] K. I. Kim., K. Jung., and J. Kim., "Face recognition using support vector machines with local correlation kernels," International Journal of Pattern Recognition and Artificial Intelligence, vol. 16 no. 1, pp. 97- 111, 2002.

[15] G.D. Guo, H.J. Zhang, and S.Z. Li., "Pairwise face recognition," in Proceedings of 8th IEEE International Conference on Computer Vision. Vancouver, Canada, July 9-12, 2001.

[16] F. Salimi., M. Sadeghi., M. S. Moin and J. Kittler., "Face Verification Using Colour Kernels," Springer-Verlag Berlin Heidelberg, pp. 522-531, 2009.

[17] M. Agarwal, N. Jain, M. Kumar and H. Agrawal, "Face Recognition Using Eigen Faces and Artificial Neural Network", International Journal of Computer Theory and Engineering, Vol. 2, No. 4, August, 1793-8201, 2010.

[18] V. More and A. Wagh, "Improved Fisher Face Approach for Human Recognition System using Facial Biometrics", International Journal of Information and Communication Technology Research, Volume 2 No. 2, February 2012.

[19] L. Chen, H. Liao, M. Ko, J. Lin and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. In IEEE Pattern Recognition 33 (10), 2000.

[20] A. Nefian and M. Hayes, " An embedded hmm-based approach for face detection and recognition," In Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 6, pp. 3553–3556, 1999.

[21] R. Kimmel and J. A. Sethian, "Computing geodesic on manifolds," in Proc. US National Academy of Science, 1998, vol. 95, pp. 8431–8435. 1998.

[22] X. Desquesnes, A. Elmoataz, O. Lézoray: "Eikonal equation adaptation on weighted graphs: fast geometric diffusion process for local and non - local image and data processing". Journal of Mathematical Imaging and Vision 46, 2 (2013), pp. 238-257, 2014.

[23] E. Carlini, M. Falcone, N. Forcadel, R. Monneau: "Convergence of a generalized fast-marching method for an eikonal equation with a velocity-changing sign". SIAM J. Numer. Anal. 46, 2920– 2952 (2008).

[24] E.W. Dijkstra, A Note on Two Problems in Connection with Graphs, Numerische Mathematik, 1 (1959), pp. 269–271, 1959.

[25] A. M.Bronstein, M. M. Bronstein, E. Gordon, R. Kimmel: "Fusion of 2D and 3D Data in Three-Dimensional Face Recognition". Image Processing, 2004. ICIP '04. 2004 International Conference on (Volume: 1), IEEE, p.p: 87 - 90. (2004).

[26] A. M.Bronstein, M. M. Bronstein, A. Spira, R. Kimmel: "Face Recognition from Facial Surface Metric", Lecture Notes in Computer Science Volume 3022, 2004, Springer Berlin Heidelberg, p.p: 225-237. (2004).

[27] R. Ahdid., S. Safi and B. Manaut, "Approach of Facial Surfaces by Contour," IEEE Xplore, International Conference Multimedia Computing and Systems (ICMCS), pp: 465-468, (2014).

[28] H. Ling and D. Jacobs. Deformation invariant image matching. In ICCV, 2005.

[29] S. Miao, H. Krim: “3D Face Recognition Based on Evolution of Iso-Geodesic Distance Curves”, Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, pp: 1134 – 1137. 2010.

[30] S. Jahanbin, H. Choi, Y. Liu, A. C. Bovik: “Three Dimensional Face Recognition Using Iso-Geodesic and Iso-Depth Curves”, Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on, pp: 1-6. 2008.

[31] L. Ballihi, A. Srivastava, B. B. Amor, M. Daoudi, D. Aboutajdine: “Which 3D geometric facial features give up your identity?”, ICB 2012, pp: 119-124. 2012.

[32] L. Ballihi, B. B. Amor, M. Daoudi, A. Srivastava, D. Aboutajdine: “Boosting 3-D-Geometric Features for Efficient Face Recognition and Gender Classification”. IEEE Transactions on Information Forensics and Security 7(6), pp: 1766-1779. 2012.

[33] E. Klassen and A. Srivastava, Geodesics Between 3D Closed Curves Using Path Straightening, Proceedings of ECCV, Lecture Notes in Computer Science, 2006, p. 95-106. Springer Berlin Heidelberg. 2006.

[34] S. Joshi, E. Klassen, A. Srivastava, and I. Jermyn: “An Efficient Representation for Computing Geodesics between n-Dimensional Elastic Shapes”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2007.

[35] A. Maalej, B. Ben Amor, M. Daoudi, A. Srivastava, S. Berretti: “Shape analysis of local facial patches for 3D facial expression recognition”. Pattern Recognition (Volume 44, Issue 8), pp: 1581-1589 (2011).

[36] R. Ahdid, S. Safi and B. Manaut: “Three Dimensional Face Surfaces Analysis using Geodesic Distance”, Journal of Computer Sciences and Applications, (Volume 3, Issue 3), pp: 67-72. (2015).