

An Intrinsic Framework for Analysis of Facial Surfaces

Chafik Samir¹, Anuj Srivastava², Mohamed Daoudi¹, Eric Klassen³

¹ GET/Telecom Lille1, LIFL (UMR USTL/CNRS 8022), France

² Department of Statistics, Florida State University, Tallahassee, FL 32306, USA

³ Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA

The date of receipt and acceptance will be inserted by the editor

Abstract A quantitative analysis of shapes of facial surfaces can play an important role in biometric authentication. The main difficulty in comparing shapes of surfaces is the lack of a canonical system to represent all surfaces. This paper overcomes that problem by proposing a specific coordinate system, on facial surfaces, that enables comparisons of geometries of faces. In this system, a facial surface is represented as a path on the space of closed curves, called facial curves, where each curve is a level curve of distance function from the tip of the nose. Defining \mathcal{H} to be the space of paths on the space of closed curves, the paper studies the differential geometry of \mathcal{H} and endows it with a Riemannian. Using numerical techniques, it computes geodesic paths between elements of \mathcal{H} that represent individual facial surfaces. This Riemannian analysis of faces is then used to: (i) find an optimal deformation from one face to another, (ii) define and compute an average face for a given set of faces, and (iii) compute distances between faces to quantify differences in their shapes. Experimental results are presented to demonstrate and support these ideas.

1 Introduction

There has been an increasing interest in recent years in analyzing shapes of 3D objects. Advances in shape estimation algorithms, 3D scanning technology, hardware-accelerated 3D graphics, and related tools, are enabling access to high quality 3D data. As technologies continue to improve, the need for automated methods for analyzing shapes of 3D objects will also grow. In terms of characterizing 3D objects, for detection, classification, and recognition, their shape is naturally an important feature. It already plays important roles in areas such as medical diagnostics, object designs, database search, and biometrics. Focusing on the last topic, our goal in this paper is to develop a framework for analyzing shapes of facial surfaces. Shape analysis involves computing metrics and “optimal deformations” between any two given objects in a manner that is invariant to certain *similarity transformations*. Typically, similarity transformations consists of the (rigid) translations and rotations, and the (non-rigid) uniform scalings of the objects involved. For the problem of shape analysis of facial surfaces, we have chosen the similarity group to consist only of translations and rotations; changes in scale of a facial surface will change the resulting deformations and the metrics. Restating our goals, we want to develop a mathematical framework for representing and comparing shapes of facial surfaces using metrics that are invariant to rotations and

translations. In this process, we will also define and compute optimal deformations of facial surfaces, from one into another. Our mathematical representations of surfaces will be such that the ensuing face comparisons are stable with respect to deformations of facial surfaces that occur during changes in facial expressions. This is an important advantage of this representation, especially when it is used for recognizing people using shapes of their facial surfaces.

What is the main difficulty in comparing shapes of two surfaces? Shape analysis should ideally be invariant to the choice of parameterizations, i.e. one should get the same comparison irrespective of parametrization of the two surfaces. To highlight this issue, consider the problem of analyzing shapes of curves in \mathbb{R}^n . In this case, there is a fixed ordering of points along the curves which helps us in parameterizing them (even though the parametrization is not unique). Therefore, one directly works with the parameterized curves and develops metrics that are invariant to re-parameterizations. For instance, if the curves are assumed to be parameterized by $s \in [0, 1]$, then the space of automorphisms of $[0, 1]$ can be identified with the space of all possible re-parameterizations of curves. (An automorphism is a smooth, invertible map from a set to itself.) So, one seeks a method to compare shapes of any two curves that is invariant to re-parametrization of one or both curves. Although seemingly difficult, this issue has been resolved by several authors [15, 11, 10, 12, 16, 21]. The key idea is that there exists an efficient computational solution, namely the dynamic programming algorithm, that can find the optimal re-parametrization for matching any two curves, at arbitrarily fine samplings. Returning to the problem of comparing facial surfaces, this problem is made difficult by the fact that there is no natural ordering of points on a surface. Therefore, the space of all possible re-parameterizations of a surface is a larger set, i.e. the automorphisms of $[0, 1]^2$, and an efficient analysis of shapes that is invariant to all re-parameterizations remains elusive. As explained later, our solution is to impose a natural hierarchy (or ordering) of points on a surface, using the idea of facial curves. This additional structure results in dividing the re-parameterizations into two smaller sets, each consisting of automorphisms of $[0, 1]$, each of which is relatively easier to solve using dynamic programming.

1.1 Current Approaches

- **3D Face Analysis:** There has been a number of papers in recent years in 3D shape matching. A common theme in this literature has been to represent facial surfaces by certain feature sets that are geometrical, such as the convex parts, areas with high curvatures, saddle points, etc [3], [2]. Although such feature definitions are intuitively meaningful, the computation of curvatures involves numerical approximation of second derivatives and is very susceptible to observation noise. Other approaches, such as those based on shape distribution [13] and conformal geometry [19], have also been proposed. Most of those works compare 3D shapes by comparing some corresponding features while our goal is to compare the facial surfaces themselves. One exception is Glaunes et al. [5] where the authors have studied diffeomorphic matching of a given pair of distributions (of points) in a general setting, with applications to various matching problems including curves, surfaces, and unlabeled points-sets.
- **Shape Analysis of Curves:** There has been a significant amount of research in analyzing shapes of open and closed curves, especially in a plane [15, 11, 10, 12, 16, 21]. The common theme has been to represent an infinite-dimensional moduli space of curves, with curves being represented in one of many possible ways, and to study the differential geometry of this space modulo the space of re-parameterizations. In case of closed curves, one uses an additional constraint to ensure closure; this results in the set of relevant curves being a nonlinear sub-manifold inside a larger Hilbert space. To compare any

two curves, one imposes a Riemannian structure on the quotient space, and computes geodesic paths under the chosen metric. The choice of a metric is important and different authors have advocated different metrics.

- **Facial Surfaces Using Facial Curves:** An important line of work in face analysis has been to use shapes of curves that lie on facial surfaces. Similar ideas have been pursued in brain surface analysis: shapes of sulcal curves have been used to study shapes of brain surfaces [8]. Curves on surfaces are usually defined using level sets of functions on these surfaces. For example, one can use the curvature tensor on a surface to obtain parabolic and ridge curves [6], or use surface gradients to extract crest lines[20]. However, computations of gradients and curvatures both involve higher order derivatives and, as mentioned above, are numerically unstable. In an attractive approach, Bronstein et al. [1] use a surface distance function to define level curves that are invariant to rigid motions. In [14], Samir et al. suggested the use of level sets of the height function (z coordinates) as curves of interest for face analysis. For comparing shapes of two facial surfaces, the basic idea is to extract level curves of the height function on each surface and to compare the corresponding curves using techniques from planar shape analysis [10]. Although the resulting metrics were reasonably successful in recognizing faces, there are some fundamental limitations with this representation. Firstly, the definition of level curves changes with the global orientation of a face and, therefore, this shape analysis is not completely invariant to rotation. Secondly, this work was intended for face recognition only and not for a Riemannian analysis of facial surfaces. In other words, it lacked a formal definition of a space of facial surfaces and a proper Riemannian structure that can lead to formal geodesic paths (or optimal deformations) between faces. Consequently, interesting statistical quantities such as the mean face are neither defined nor computed.

1.2 Our Approach

Our approach is to represent facial surfaces as an indexed collection of closed curves on faces, termed facial curves, and to apply tools from shape analysis of curves. This paper is an **extension of the framework** introduced in [14] in two ways: (i) the representations, metrics, and analysis is now completely invariant to rigid rotations and translations of the facial surfaces, and (ii) there is a formal Riemannian analysis on space representing facial surfaces, with precise mathematical definitions for spaces, metrics, geodesics, and statistics of facial surfaces. The representation of a facial surface is based on facial curves, as in [14], but the definition of facial curves is now different. Facial curves are defined here as level curves of a *surface distance function*. This function, at a point on the surface, is defined to be the length of the shortest path between that point and a fixed reference point (taken to be the tip of the nose) along the facial surface. Shapes of level curves of this function are invariant to rotation and translation of the facial surface. Visually these curves seem to be full of facial features that can prove to be important in face classification and recognition. Bronstein et al. [1] have argued that these level curves are stable with respect to changes in facial expressions and, hence, are useful in classification. These are closed curves in \mathbb{R}^3 while the curves used in [14] are planar. An indexed collection of such curves will be used to mathematically represent the corresponding facial surface. Denoting this indexed collection as a parameterized path on the space of closed curves in \mathbb{R}^3 , we formalize a mathematical representation of facial surfaces and focus on the space of all such representations. Choosing a Riemannian metric, we endow this space with a Riemannian structure and derive an algorithm for computing geodesic paths between given facial surfaces on this space. This tool for computing geodesic is shown to be useful in finding optimal deformations between faces, for registering points of facial surfaces, and for computing “average faces”.

Table 1 List of symbols and their definitions used in this paper.

Symbol	Definition /Explanation
S	an observed facial surface
$dist$	length of the shortest path on S between any two points
λ	a variable for the value of $dist$
$c_\lambda(s)$	level curve of $dist$ on S at the level λ , parameterized by s such that $\frac{dc_\lambda}{ds}(s) = \text{constant}$.
l_λ	the length of the curve c_λ .
p_λ	the starting point of the curve c_λ , $p_\lambda = c_\lambda(0)$
$v_\lambda(s)$	the scaled velocity vector, $v_\lambda(s) = \frac{1}{l_\lambda} \frac{dc_\lambda}{ds}(s)$
v_0	the velocity function of a unit circle in $X - Y$ plane, $v_0(s) = s$
(a, b)	the Euclidean inner product in \mathbb{R}^3
\mathcal{C}	the set of all closed curves in \mathbb{R}^3 , $\mathcal{C} = \{v v : [0, 1] \mapsto \mathbb{S}^2, \int_0^1 v(s) ds = 0, (v(s), v(s)) = 1, \forall s\}$
$T_v \mathcal{C}$	the space of all tangents to \mathcal{C} at v It is given by $\{f f : [0, 2\pi] \mapsto \mathbb{R}^3, \forall s, (f(s), v(s)) = 0\}$
$\langle f_1, f_2 \rangle$	the Riemannian metric on \mathcal{C} , $\int_0^{2\pi} f_1(s) f_2(s) ds$
$\phi_t(v_1, v_2)$	a geodesic path in \mathcal{C} , from v_1 to v_2 , parameterized by $t \in I \equiv [0, 1]$ $\phi_0(v_1, v_2) = v_1, \phi_1(v_1, v_2) = v_2$
α	an indexed collection of closed curves in \mathbb{R}^3 $\alpha : [0, L] \mapsto \mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+, \alpha(\lambda) = (p_\lambda, v_\lambda, l_\lambda)$
\mathcal{H}	the space of all parameterized paths in $\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+$ $\mathcal{H} = \{\alpha : [0, L] \mapsto (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+), \alpha(0) = (r, v_0, 0)\}$
$T_\alpha \mathcal{H}$	the tangent space of \mathcal{H} at α $\{(u, w, x) u(\lambda) \in \mathbb{R}^3, w(\lambda) \in T_{\alpha(\lambda)} \mathcal{C}, x(\lambda) \in \mathbb{R}\}$
$\langle\langle (u_1, w_1, x_1), (u_2, w_2, x_2) \rangle\rangle$	Riemannian metric on \mathcal{H} $= \int_0^L ((u_1(\lambda), u_2(\lambda)) + \langle w_1(\lambda), w_2(\lambda) \rangle + x_1(\lambda)x_2(\lambda)) d\lambda$
\mathcal{P}	the space of all paths β in \mathcal{H}
$\psi_t(\alpha_1, \alpha_2)$	the geodesic path in \mathcal{H} from α_1 to α_2 parameterized by t .

The rest of this paper is organized as follows. A mathematical representation of a facial surface is introduced in Section 2, with examples on how to generate these representations. Towards the goal of imposing Riemannian structures, Section 3 studies the structure of the space of facial curves and Section 4 studies a similar structure for the space of facial surfaces.

2 Mathematical Representation of a Facial Surface

Consider a facial surface S to be a two-dimensional smooth connected manifold with a genus zero. We assume that holes on S associated with eyes, nose, and mouth have already been patched smoothly. In practice S will be a triangulated mesh with a collection of connected edges and vertices, but we start by assuming that it is a continuous surface. One can view it as a smooth embedding of the upper unit hemisphere \mathbb{S}_+^2 in \mathbb{R}^3 . Some pictorial examples of S are shown in Figure 1 where 2D images and their corresponding facial surfaces associated with six facial expressions (neutral, smile, frown, angry, squint, scared) of the same person are displayed.

As mentioned earlier, the biggest challenge in analyzing the shape of S is to represent it in a way that allows for efficient comparison of faces. Our approach is to represent a facial surface as an indexed collection of curves, or as a parameterized path in the space of closed curves; that allows the breaking of a 2D registration problem into two independent 1D registration problems, as described next. Let r be a prominent reference point on S that

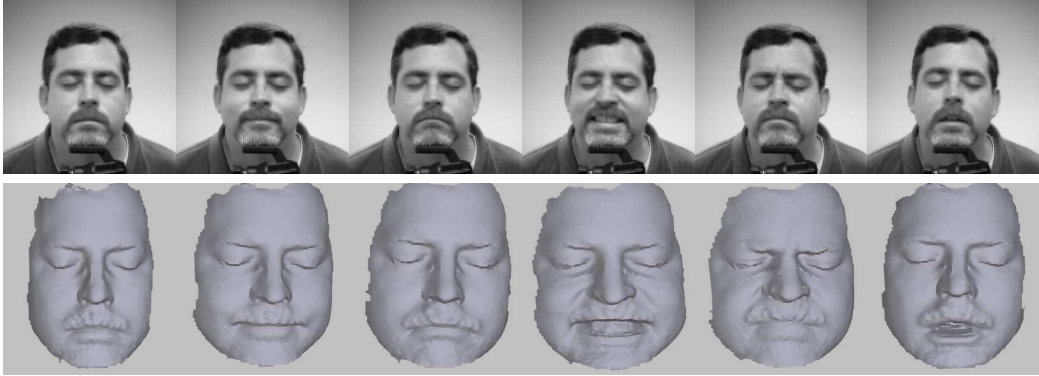


Fig. 1 2D images and their corresponding facial surfaces of same person under different facial expressions: neutral, smile, frown, angry, squint, and scared.

is easily detectable. In this paper, we have chosen r as the tip of the nose. Then, define a function $dist$ to be the **surface distance function** from r to any point on the facial surface. In other words, $dist(r, q)$ is the length of the shortest path connecting r and q while staying on S . Using this function, one can define facial curves as the level sets of $dist$:

$$c_\lambda = \{q \in S \mid dist(r, q) = \lambda\} \subset S, \lambda \in [0, \infty). \quad (1)$$

For large values of λ , the set c_λ can be expected to be empty. On the other hand, for $\lambda = 0$, the set c_λ is the singleton $\{r\}$. For values of λ in between those two extremes, we expect c_λ to capture of certain local geometry of the surface S . So far c_λ is just a collection of points but they can be ordered easily and we consider $c_\lambda(s)$ as a parameterized, closed curve that is parameterized by constant speed over the interval $[0, 1]$, i.e. $\frac{dc_\lambda}{ds}(s) = \text{constant}$. Three sets of variables describe this curve:

- **Initial Position:** We will denote the starting point on this curve by $p_\lambda \in \mathbb{R}^3$, i.e. $c_\lambda(0) = p_\lambda$.
- **Length:** Let l_λ denote the length of the curve c_λ .
- **Direction function:** The unit velocity vector $v_\lambda(s) \equiv \frac{1}{\|\frac{dc_\lambda}{ds}(s)\|} \frac{dc_\lambda}{ds}(s)$. Note that $v_\lambda(s) \in \mathbb{S}^2$.

We will represent the original curve $c_\lambda(s)$ by the triple $(p_\lambda, v_\lambda, l_\lambda)$. For $\lambda = 0$, c_λ collapses into the point r and we will use the convention that $c_0(s) = (r, v_0(s), 0)$ where $v_0(s) = [-\sin(s) \cos(s) 0]'$ is the velocity function of a unit circle in the $X - Y$ plane. We will choose an upper bound L of λ so that c_λ for $\lambda \leq L$ is a continuous, smooth, closed curve in \mathbb{R}^3 . This helps in cropping some erroneous and unreliable data at the boundaries.

Next we need to specify the space in which these representations take values. Define a set:

$$\mathcal{C} = \{v : [0, 1] \mapsto \mathbb{S}^2 \mid \int_0^1 v(s) ds = 0, (v(s), v(s)) = 1, \forall s\}.$$

\mathcal{C} is the set of all velocity functions that correspond to constant-speed, closed curves in \mathbb{R}^3 . For a given triple (p, v, l) , one can re-construct the curve using $c(\tau) = p + l \int_0^\tau v(s) ds$.

Before we proceed, we illustrate some computational steps needed to extract c_λ from a given observation of S . We summarize the extraction and the parametrization of facial curves in the following algorithm:

Algorithm 1 : Facial curve extraction and parametrization

Given a facial surface S , the c_λ s are determined as follows:

- 1- Preprocess the surface according to Section 2.1.
 - 2- Calculate $dist$ value on all the vertices of the mesh according to Section 2.2.
 - 3- Crop the mesh at $dist$ value L .
 - 4- Extract the level set of $dist$ (collection of unordered points) at the level $\lambda \in [0, L]$ according to Section 2.3.
 - 5- Order these points and smooth the resulting curve c_λ .
 - 6- Uniformly resample c_λ .
 - 7- Compute the tangent vector function $v_\lambda(s)$ and the curve length l_λ ; and set the parametrization $c_\lambda = (p_\lambda, v_\lambda, l_\lambda)$.
-

2.1 Surface Pre-Processing

The extraction part involves several steps to ensure robustness. Firstly, we remove disconnected parts so that the surface distance function $dist$ is well defined. Secondly, we fill holes by applying a linear interpolation, and then make the meshes regular by fixing the length of all edges equal to a standard value, resulting in a Delaunay triangulation. Finally, we increase the number of vertices while keeping the mesh regular so that the surface is smooth and the resolution is finer. Shown in the top row of Figure 2 is an illustration of preprocessing a facial surface. Note that the hole on the nose was filled and the mesh has been regularized. Shown in the bottom row are examples of refining a non-uniform mesh into a dense, uniform mesh.

2.2 Surface Distance Function

Once the facial surface is pre-processed, the tip of the nose can be easily detected. Denote this point as $r \in \mathbb{R}^3$. A surface distance function $dist : S \rightarrow \mathbb{R}_+$ is then defined as the length of the shortest path (on S) from r to any vertex q in the mesh. This function has been used previously in many surface matching methods. For instance, Hilaga [7] used this function to construct a Reeb graph and Bronstein et al. [1] used it to crop facial surfaces at their boundaries. Several methods have been proposed to compute $dist$ on S ; a summary and comparison of those methods is presented in [18]. We will use the Dijkstra's algorithm [4] to compute $dist$ that has a computational cost of $O(n)$ with n being the number of vertices in the mesh. The resulting $dist$ is invariant to rigid rotations and translations. As verified experimentally later, it is also relatively robust to the deformations resulting from changes in facial expressions.

2.3 Facial Curves Extraction

For a $\lambda \in [0, L]$, we seek the level set c_λ in a given surface S . For practical purposes, we modify the definition of c_λ to consist of all the points whose distance $dist$ from r is in $[\lambda - \delta, \lambda + \delta]$, where δ is a pre-determined error value. A smaller value of δ ensures continuity and smoothness of the extracted curves. Shown in Figure 3 are examples of extracted facial curves corresponding to two different value of δ , the left picture is for a larger δ while the right one is for a smaller δ . Figure 4 shows more examples of extracted facial curves. The top row shows some facial curves drawn on the facial surface while the bottom row shows increasing number of curves extracted from a facial surface.

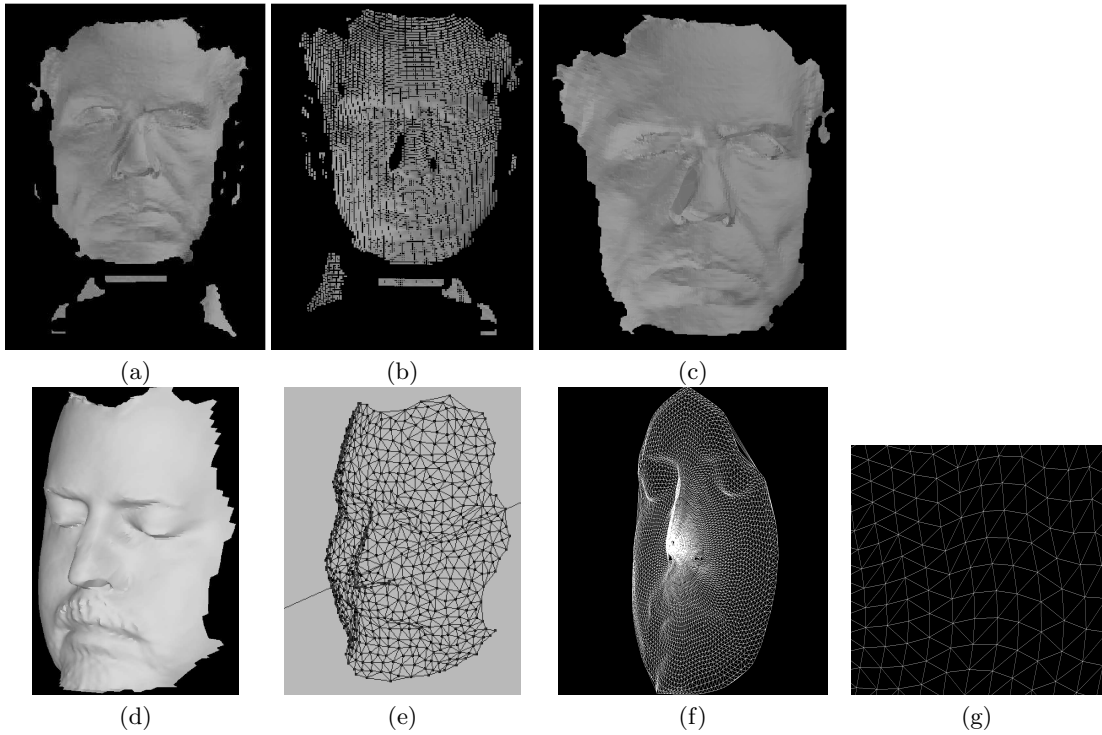


Fig. 2 (a), (b): An example of facial surface before pre-processing and (c) same surface after pre-processing. (d) A facial surface, (e) its corresponding nonuniform mesh before regularization, (f) uniform mesh after regularization, (g) a part of that mesh enlarged to show uniformity.



Fig. 3 Left: extracted level set on an irregular mesh. Right: an extracted level set on the regular mesh after refinement.

2.4 Representations Using Facial Curves

Now we are ready to state our mathematical representation of a facial surface. Consider a path $\alpha : [0, L] \mapsto (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+)$, such that $\alpha(\lambda) = (p_\lambda, v_\lambda, l_\lambda)$ for $\lambda \in [0, L]$. Additionally, fix the starting point of α to be $(r, v_0, 0)$. Let \mathcal{H} be the set of all such paths, i.e.

$$\mathcal{H} = \{ \alpha : [0, L] \mapsto (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+) | \alpha(0) = (r, v_0, 0) \} . \quad (2)$$

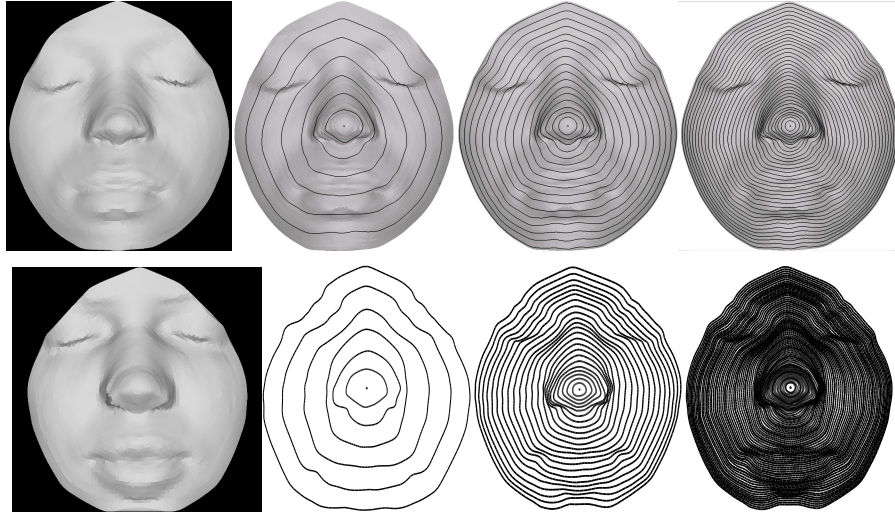


Fig. 4 Top: An example of facial surface (left), an extracted level set curve drawn on the surface. Bottom: Increasing number of facial curves extracted from the surface in the left panel.

We will represent facial surfaces and analyze them as elements of \mathcal{H} . Shown in Figure 5 are examples of a facial surface represented using a large number of facial curves. The top picture shows the original surface S while the bottom row shows a rendering of $c_{\lambda S}$ for an increasing number of λ s as we go from left to right. From an implementation point of view, one has to settle for a finite number of curves in representing S . The actual choice of which and how many curves are used in the analysis will ultimately depend on the application.

We note the following properties of this representations:

1. **Invariance to Rotation and Translation:** A good representation of S should be invariant to translation and rotation of S . The choice of the surface distance function $dist$ to define facial curves ensures the invariance of level curves c_{λ} . Since α is simply a collection of facial curves $c_{\lambda S}$, it is also invariant to these transformations.
2. **Surface Reconstruction:** In representing S by α , have we lost any information about S ? For an arbitrary point $p \in S$, there exists a λ such that p lies on the curve c_{λ} . Therefore, all points on S are present in α on one level curve or another. The facial curves $c_{\lambda S}$ are non-overlapping and, hence, no point p is represented twice in α . Thus, each point p is present once and only once in the set α . Therefore, considered as a collection of points S and α are identical. Shown in Figure 5 are some examples of α associated with several faces. Each face in this figure is of the same person but at different facial expression. Can we reconstruct a facial surface using its representation $\alpha \in \mathcal{H}$? To reconstruct, we first generate the Delaunay triangulation of the set of points contained in α and, then, use this triangulation to render a surface. An example is shown in Figure 6 which pictorially shows an α or a collection of facial curves (left), the corresponding mesh generated using a delaunay triangulation (middle), and the resulting smooth facial surface (right).

We have represented facial surfaces as elements of \mathcal{H} . The next step is to impose a Riemannian structure on \mathcal{H} and to compute geodesic paths between observed facial surfaces. Towards that goal, we first study the differential geometry of \mathcal{C} and present algorithms for computing geodesics between given facial curves.

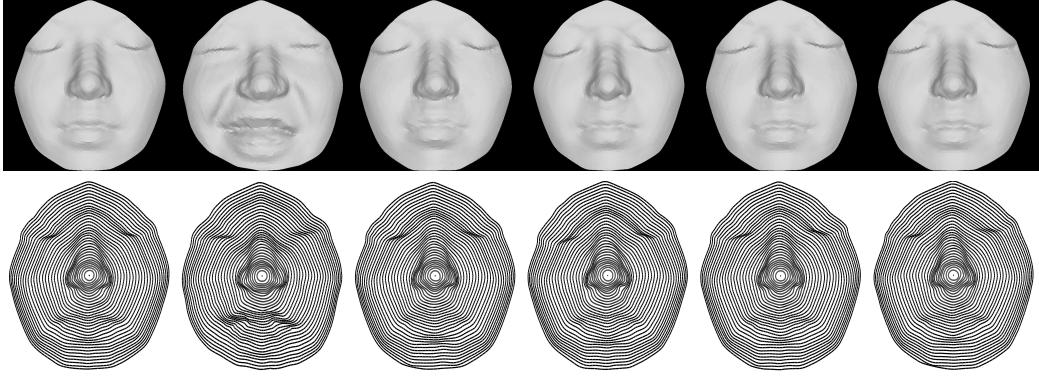


Fig. 5 Facial surfaces of the same person under different facial expressions, their correspondents representations in \mathcal{H} .

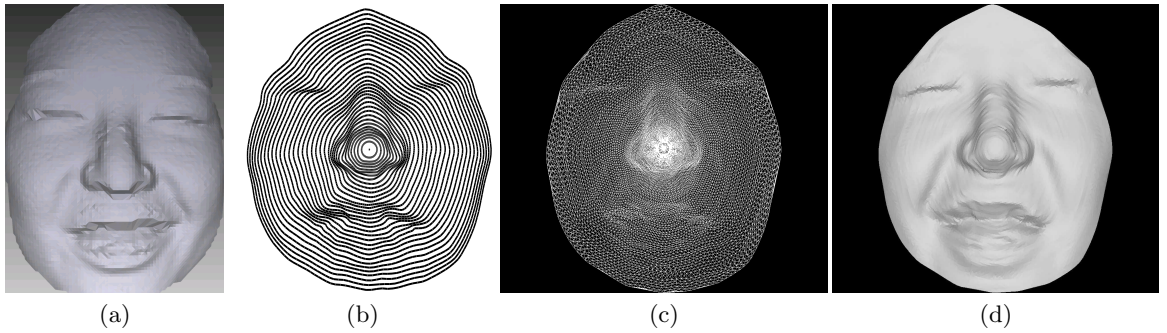


Fig. 6 Reconstruction example: (a) an original facial surface, (b) its representation as a large collection of curves in \mathcal{H} , (c) a dense triangulated mesh formed from curves, and (d) the reconstructed surface.

3 Riemannian Analysis of Facial Curves

A facial curve defined in the previous section is represented by: (i) a starting point $p \in \mathbb{R}^3$, (ii) a velocity function $v \in \mathcal{C}$, and (iii) a length $l \in \mathbb{R}_+$. Given two such facial curves, we address the question: How to form a geodesic path between them in an appropriate space of curves under a chosen Riemannian metric? While the parts dealing with the starting points and the lengths are simple, the main difficulty lies in computing a geodesic path between the velocity functions of the two given curves in \mathcal{C} . This exact problem has been solved in Klassen et al. [9], and we will adapt their solution to the study of facial curves. For the benefit of readers, the main idea behind that approach has been summarized in this section.

To develop a geometric framework for analyzing elements of \mathcal{C} , it is important understand its tangent bundle and to impose a Riemannian structure on it. Recall that $v(s) \in \mathbb{S}^2$ and, consequently, v is a curve on \mathbb{S}^2 ; a vector f tangent to \mathcal{C} at v can also be viewed as a field of vectors tangent to \mathbb{S}^2 along v . The space of all such tangent vectors, denoted by $T_v(\mathcal{C})$, is given by:

$$T_v(\mathcal{C}) = \{f | f : [0, 1] \mapsto \mathbb{R}^3, \forall s \ (f(s) \cdot v(s)) = 0, \int_0^1 f(s) ds = 0 \}. \quad (3)$$

To impose a Riemannian structure on \mathcal{C} , we will assume the following inner product on $T_v(\mathcal{C})$: for $f, g \in T_v(\mathcal{C})$,

$$\langle f, g \rangle = \int_0^1 (f(s) \cdot g(s)) ds . \quad (4)$$

For any two closed curves, denoted by v_0 and v_1 in \mathcal{C} , we are interested in finding a geodesic path between them in \mathcal{C} under the metric given in Eqn. 3. This is accomplished using a *path-straightening* approach presented in [9]. The basic idea is to start with any path $\phi(t)$ connecting v_0 and v_1 . That is $\phi : [0, 1] \mapsto \mathcal{C}$ such that $\phi(0) = v_0$ and $\phi(1) = v_1$. Then, one iteratively “straightens” ϕ until it achieves a local minimum of the energy:

$$E_c(\phi) \equiv \frac{1}{2} \int_0^1 \left\langle \frac{d\phi}{dt}(t), \frac{d\phi}{dt}(t) \right\rangle dt , \quad (5)$$

over all paths from v_0 to v_1 . The iteration is performed using a gradient approach, i.e. update ϕ iteratively in the direction of negative gradient of E till the gradient becomes zero. It has been shown in [17] that a critical point of E is a geodesic on \mathcal{C} . Shown in Figure 7 is an example of the geodesic path between two facial curves and the evolution of E_c during path straightening.

Remark: One major difference between the framework in [9] and here is that the starting point on the individual facial curves are kept fixed here. In [9], the variability due to different placements of starting points, on a closed curve, was modeled using the action of unit circle \mathbb{S}^1 on \mathcal{C} and the geodesics were actually computed in the quotient space \mathcal{C}/\mathbb{S}^1 . However, in the current paper, we use the vertical plane containing the bridge of the nose to determine the starting points on each facial curve, and we fix them. Hence, the geodesics here are computed in \mathcal{C} and not in the quotient space \mathcal{C}/\mathbb{S}^1 . In computational terms, this involves less work than was done to find geodesics in [9].

4 Riemannian Analysis of Faces on \mathcal{H}

Next, we describe our approach to construct an “optimal” deformation from one facial surface to another. Since facial surfaces are represented as elements of \mathcal{H} , a natural formulation of “optimal” is to consider the two corresponding elements in \mathcal{H} and to construct a geodesic path connecting them in \mathcal{H} . The definition of geodesic is with respect to a Riemannian metric that measures bending of facial surfaces. To actually compute geodesics, we need to study the differential geometry of \mathcal{H} and to explain our choice of a Riemannian structure on it. The tangent space of \mathcal{H} is given by:

$$T_\alpha(\mathcal{H}) = \{(u, w, x) \mid \forall \lambda \in [0, L], \quad u(\lambda) \in \mathbb{R}^3, \quad w(\lambda) \in T_{v_\lambda}(\mathcal{C}), \quad x(\lambda) \in \mathbb{R}\} , \quad (6)$$

where $T_{v_\lambda}(\mathcal{C})$ is as specified in Eqn. 3. \mathcal{H} is an infinite non linear manifold, and to impose a Riemannian structure on it, we define the Riemannian metric as follows:

$$\langle (u_1, w_1, x_1), (u_2, w_2, x_2) \rangle = \int_0^L ((u_1(\lambda), u_2(\lambda)) + \langle w_1(\lambda), w_2(\lambda) \rangle + x_1(\lambda)x_2(\lambda)) d\lambda . \quad (7)$$

The inner product on tangent spaces of \mathcal{C} is defined in Eqn. 4.

Let S_0 and S_1 be any two given facial surfaces, and α_0 and α_1 be the corresponding elements in \mathcal{H} , respectively. Our goal is to construct a geodesic path $\Psi(t)$ in \mathcal{H} , parameterized by time t , such $\Psi(0) = \alpha_0$ and $\Psi(1) = \alpha_1$. For each $\lambda \in [0, L]$, we have

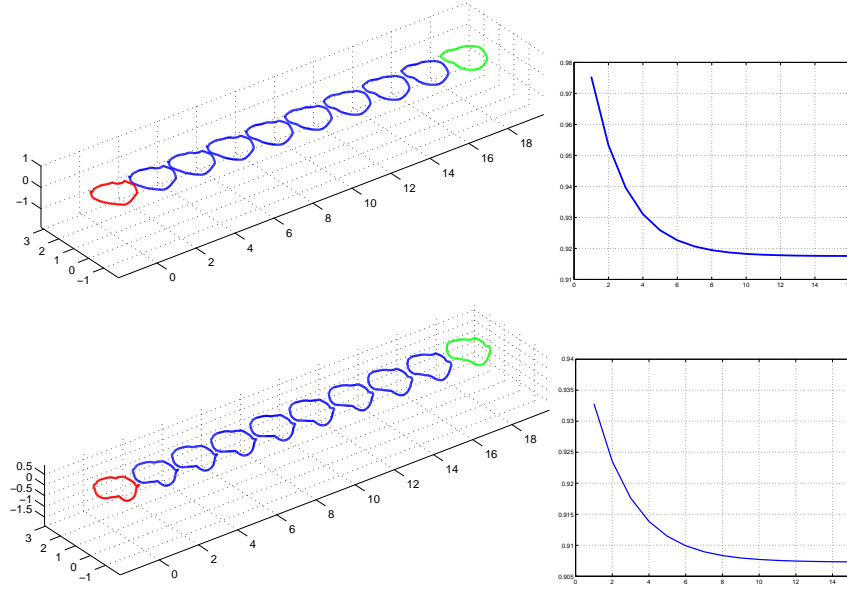


Fig. 7 The geodesic path between two facial curves c_λ^1 and c_λ^2 , and the corresponding evolution of the energy E_c .

$\alpha_i(\lambda) = (p_{i,\lambda}, v_{i,\lambda}, l_{i,\lambda}) \in (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+)$. For each of the components we compute the geodesic paths independently by defining:

$$\Psi(t) = \{\Psi_\lambda(t), \lambda \in [0, L]\}, \quad \text{where } \Psi_\lambda(t) = (\Psi_\lambda^p(t), \Psi_\lambda^v(t), \Psi_\lambda^l(t)). \quad (8)$$

Ψ_λ is the geodesic path between the corresponding facial curves; it is composed of three components. Ψ_λ^p is the geodesic on the initial position component, Ψ_λ^v is over the shape component, and Ψ_λ^l is over the length component. These individual components are computed as follows:

1. **Shape of Curves:** Let $\Psi_\lambda^v(t)$ be a geodesic path in \mathcal{C} , constructed using path straightening [9], such that $\Psi_\lambda^v(0) = v_{0,\lambda}$ and $\Psi_\lambda^v(1) = v_{1,\lambda}$. This is accomplished as described in Section 3.
2. **Initial Position:** The geodesic path between the starting points $p_{0,\lambda}$ and $p_{1,\lambda}$ in \mathbb{R}^3 is given by a straight line in \mathbb{R}^3 . That is, $\Psi_\lambda^p : [0, 1] \rightarrow \mathbb{R}^3, t \mapsto tp_{1,\lambda} + (1-t)p_{0,\lambda}$.
3. **Length:** The geodesic path between the lengths $l_{0,\lambda}$ and $l_{1,\lambda}$ in \mathbb{R} is given by a straight line in \mathbb{R}^+ . That is, $\Psi_\lambda^l : [0, 1] \rightarrow \mathbb{R}^+, t \mapsto tl_{1,\lambda} + (1-t)l_{0,\lambda}$.

Proposition 1 *The path Ψ defined in Eqn. 8 is a geodesic in \mathcal{H} , such that $\Psi(0) = \alpha_0$ and $\Psi(1) = \alpha_1$.*

Proof: For a fixed λ , the three components $(\Psi_\lambda^p, \Psi_\lambda^v, \Psi_\lambda^l)$ are individually geodesics in spaces \mathbb{R}^3 , \mathcal{C} , and \mathbb{R}_+ , respectively, by construction. Thus, Ψ_λ is a geodesic in the product space with respect to the metric given in Eqn. 7. As shown in Appendix, the full set of such geodesics Ψ is a geodesic if each one of Ψ_λ is a geodesic.

Next, we summarize the main steps required in computation of a geodesic path between given two facial surfaces.

We performed a number of experiments using facial surfaces of same and different persons under a variety of facial expressions, and studied the resulting geodesic paths. Some examples are shown in Figures 8 - 11. Figures 8(a),(b) show facial surfaces S_1 and S_2 of the same

Given two different surfaces S_1 and S_2 , this algorithm summarizes computational steps to construct a geodesic path between their representations in \mathcal{H} .

- 1- Extract level curves c_λ^1 and c_λ^2 , for each λ , using Algorithm 1.
 - 2- Represent each surface as an element of \mathcal{H} and denote them by α_0 and α_1 , respectively.
 - 3- Compute the geodesics $\Psi_\lambda(t)$, at each level $\lambda \in [0, L]$, in $\mathbb{R} \times \mathcal{C} \times \mathbb{R}_+$.
 - 4- Combine these geodesic paths to form a path $\Psi(t)$ in \mathcal{H} .
 - 5- Reconstruct a facial surface for finitely spaced samples of $t \in [0, 1]$, using the collection $\{\Psi_\lambda(t), \lambda \in [0, L]\}$.
-

person under neutral expression and smiling, respectively. Figure 8(c) shows the geodesic path between S_1 (shown in far left) and S_2 (shown in far right). Drawn in between are facial surfaces denoting equally spaced points along the geodesic path. In terms of the Riemannian metric chosen, these paths denote the optimal deformations in going from the first face to the second and the path lengths quantify the amount of deformations. In Figure 9(a), we show two examples of geodesic paths between facial surfaces of the same person under different facial expressions. In Figure 9(b), we show two examples of geodesic paths between facial surfaces of the same different people. Figures 9(a) show geodesic path between the same person under different expressions, while the figures 9(c) show geodesic paths between different persons. Each of the path has been shown from two different views.

To give more details about the deformation of facial surfaces, figures 9(b,d) display a path of facial surfaces from different viewpoints, it can be shown in 9(b) that strong deformations are obtained near the mouth that start to be close and then slowly opened during the path. However, the figure 9(d) shows strong deformations near eyes, mouth, nose and chin. Figures 10 and 11 show many more examples of these geodesic paths.

Experiments show that the proposed geodesic distance is relatively stable to changes in facial expressions. That is, the deformation caused by changes in expressions is much smaller in terms of the geodesic distances, when compared to the distances between faces of different people. Shown in Figure ?? are two examples of such experiments. Each of 12×12 the matrix shows pairwise distances between 12 facial surfaces, where the first six belong to the same person (under different expressions) and the next six belong to six different persons. The resulting distances are shown as a matrix in Figure ?. In this visualization of the matrix, the lightness of each element ($i;$) is proportional to the magnitude of the distances between faces i and j . That is, each row and column represent the distances for a face when compared to all 12 faces. Darker elements represent better matches, while lighter elements indicate worse matches. The larger values of distances (lighter elements) involving the last six faces confirms that geodesic distances are smaller when the faces of the same person are involved.

5 Karcher Means of Facial Surfaces

For future statistical analysis, we are interested in defining a notion of “mean” for a given set of facial surfaces. The Riemannian structure defined on \mathcal{H} enables us to perform such statistical analysis for computing faces mean and variance. There are at least two ways of defining a mean value for a random variable that takes values on a nonlinear manifold. The first definition, called *extrinsic mean*, involves embedding the manifold in a larger vector space, computing the Euclidean mean in that space, and then projecting it down to the manifold. The other definition, called the *intrinsic mean* or the *Karcher mean* utilizes the intrinsic geometry of the manifold to define and compute a mean on that manifold. It is defined as follows: Let $d(S_i, S_j)$ denote the length of the geodesic from S_i to S_j in \mathcal{H} . To

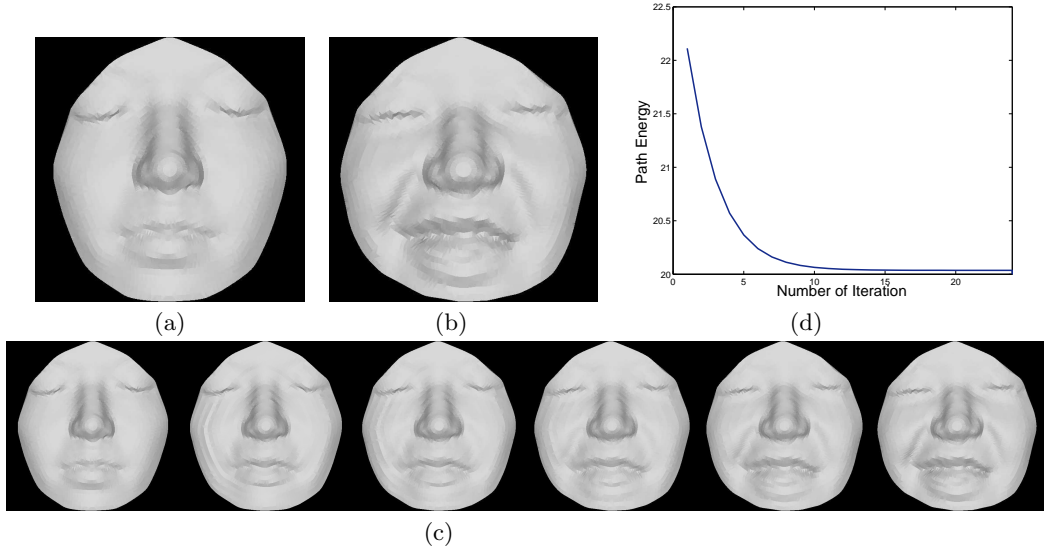


Fig. 8 (a) and (b): Facial surfaces S_1 and S_2 of the same person under neutral and smile, respectively. (c): The geodesic paths between S_1 far left and S_2 far right; drawn in between are facial surfaces denoting equally spaced point along this geodesic path. (d): The evolution of the path energy as a function of straightening iterations.

calculate the Karcher mean of facial surfaces $\{S_1, \dots, S_n\}$ in \mathcal{H} , define the variance function:

$$\mathcal{V} : \mathcal{H} \rightarrow \mathbb{R}, \mathcal{V}(S) = \sum_{i=1}^n d(S_i, S)^2 \quad (9)$$

The Karcher mean is then defined by:

$$\bar{S} = \arg \min_{\mu \in \mathcal{H}} (\mathcal{V}(\mu)) \quad (10)$$

Since \mathcal{H} is complete, the intrinsic mean as defined above always exist. However, it may not be unique, i.e. there may be a set of points in \mathcal{H} for which the minimizer of \mathcal{V} is obtained. To interpret geometrically, \bar{S} is an element of \mathcal{H} , that has the smallest deformation from all the given facial shapes.

We present a commonly used algorithm for finding Karcher mean for a given set of facial surfaces. This algorithm uses the gradient of \mathcal{V} , in the space $T_\mu(\mathcal{H})$, to update the current mean μ . If the points $\{S_1, \dots, S_n\}$ are clustered fairly close together it has been proven that

Algorithm 2 : Gradient search

Set $k = 0$. Choose some time increment $\epsilon \leq \frac{1}{n}$. Choose a point $\mu_0 \in \mathcal{H}$ as an initial guess of the mean. (For example, one could just take $\mu_0 = S_1$.)

1- For each $i = 1, \dots, n$ choose the tangent vector $f_i \in T_{\mu_k}(\mathcal{H})$ which is tangent to the geodesic from μ_k to S_i , and whose norm is equal to the length of this shortest geodesic. The vector $g = \sum_{i=1}^n f_i$ is proportional to the gradient at μ_k of the function \mathcal{V} .

2- Flow for time ϵ along the geodesic which starts at μ_k and has velocity vector g . Call the point where you end up μ_{k+1} , i.e. $\mu_{k+1} = \Psi(\mu_k; \epsilon; g)$.

3- Set $k = k + 1$ and go to step 1.

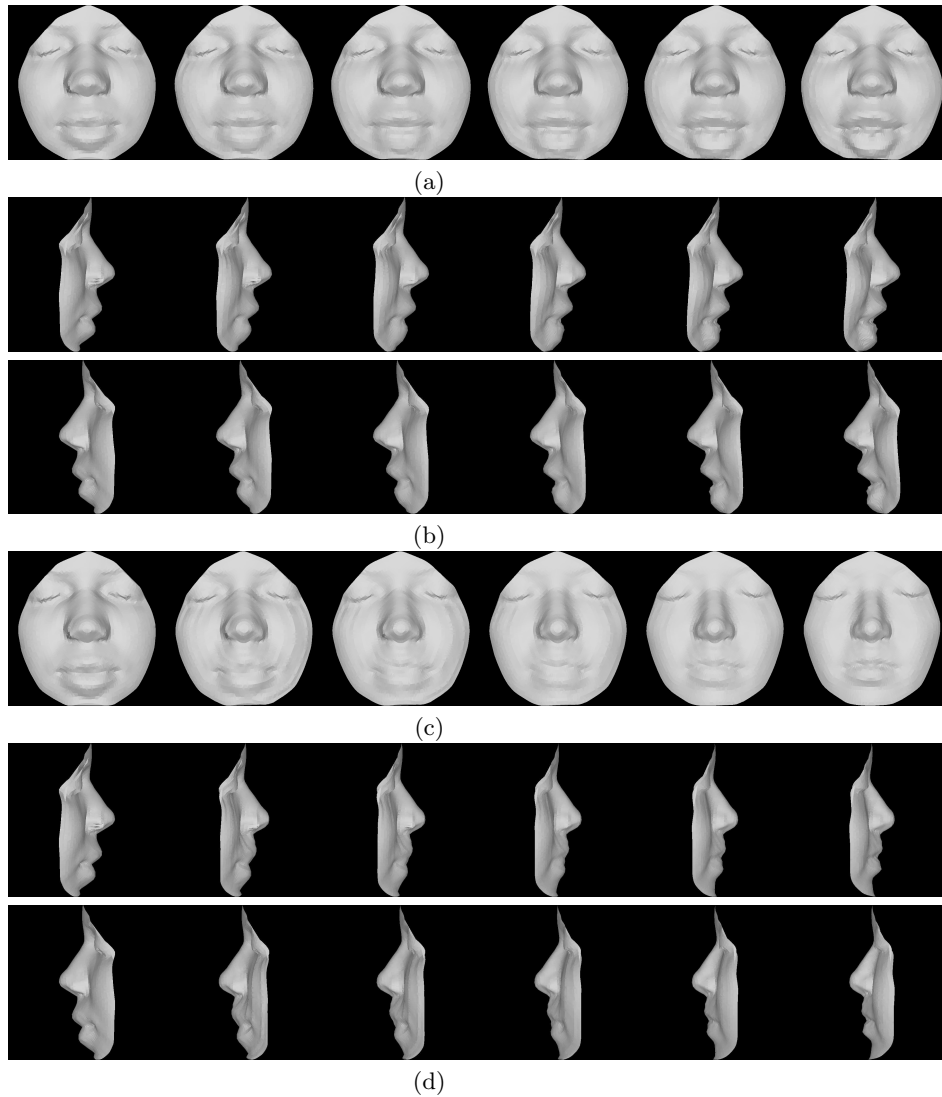


Fig. 9 Geodesic Paths between: (a) same person under different facial expressions, viewed from different viewpoints in (b). Different persons in (c) viewed from different viewpoints in (d).

the Karcher mean exists, is unique, and can be found by the above algorithm.

An example of using Karcher mean to compute the average face is shown in Figure 14. Here the sample set consists of six facial surfaces of the same person under different facial expressions, and the mean face is shown in the far right panel. We have magnified the display of Karcher mean to study the features retained by this surfaces from the original set. Figure 15 shows the Karcher mean face for six facial surfaces of different persons. Note that the facial surfaces of the same person, under different facial expressions, which are close to each other in terms of geodesic distances, are quite close to their Karcher mean. For a clustering or an identification purposes, they could be accurately been represented by their Karcher mean. However, the Karcher mean of faces of different people is far, in term of geodesic distance, to each of the faces, and appears to be a blurred version of the individual faces.

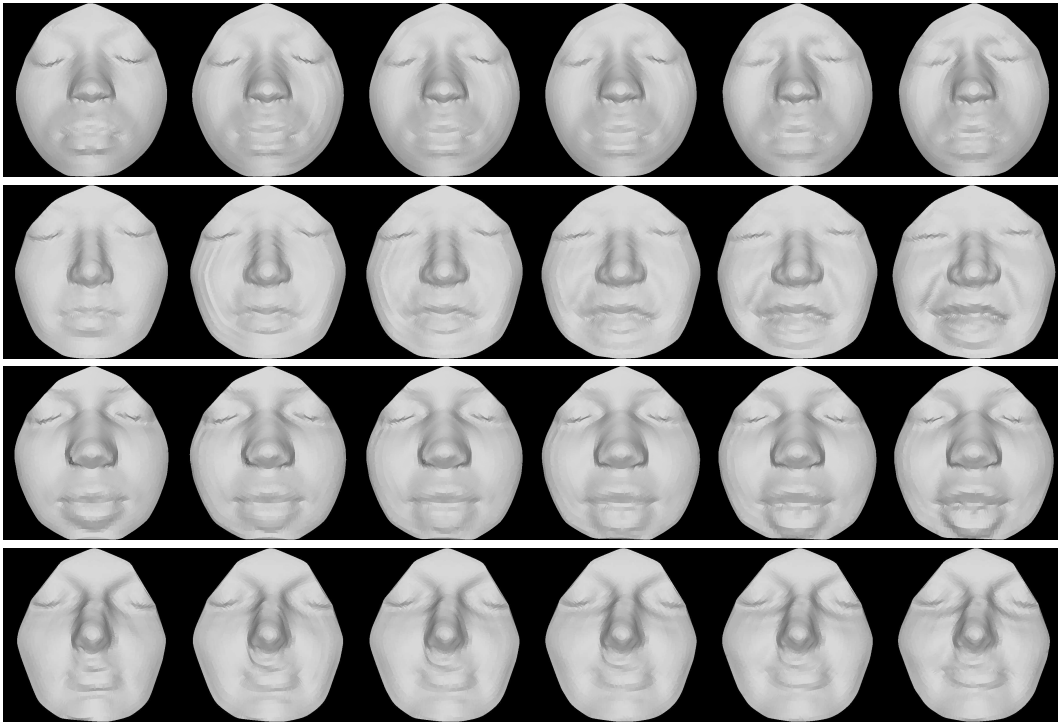


Fig. 10 Geodesic Paths between facial surfaces of same people under different expressions.

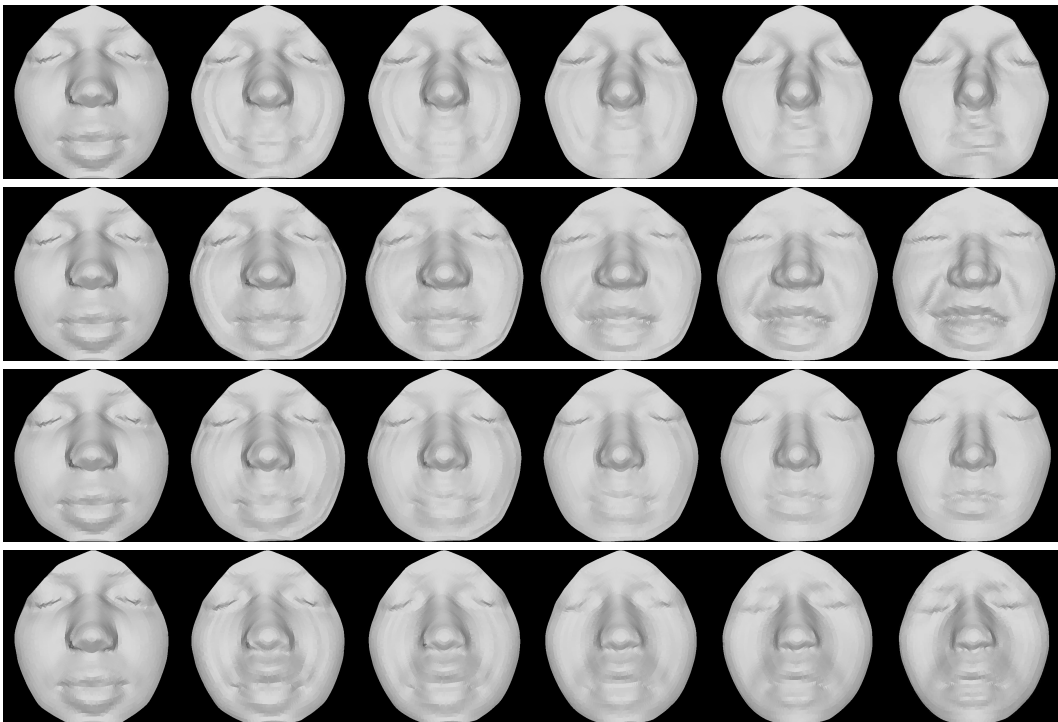


Fig. 11 Geodesic Paths between facial surfaces of different people.

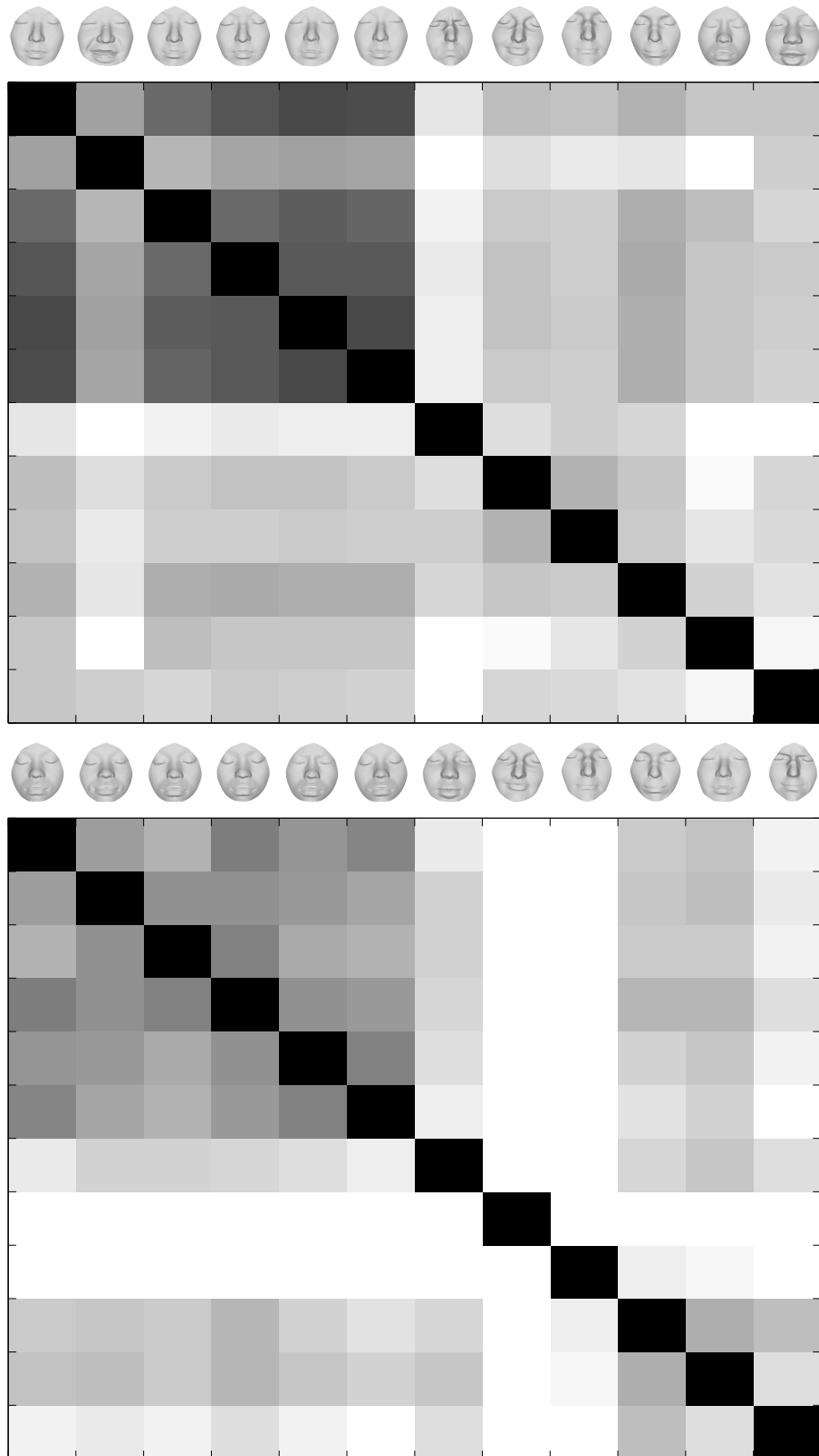


Fig. 12 Matrix of pairwise distances between 12 facial surfaces in \mathcal{H} . The first six belong to the same person, while the next six belong to six different persons. This matrix shows the stability of our geodesic distance with respect to changes in facial expressions.

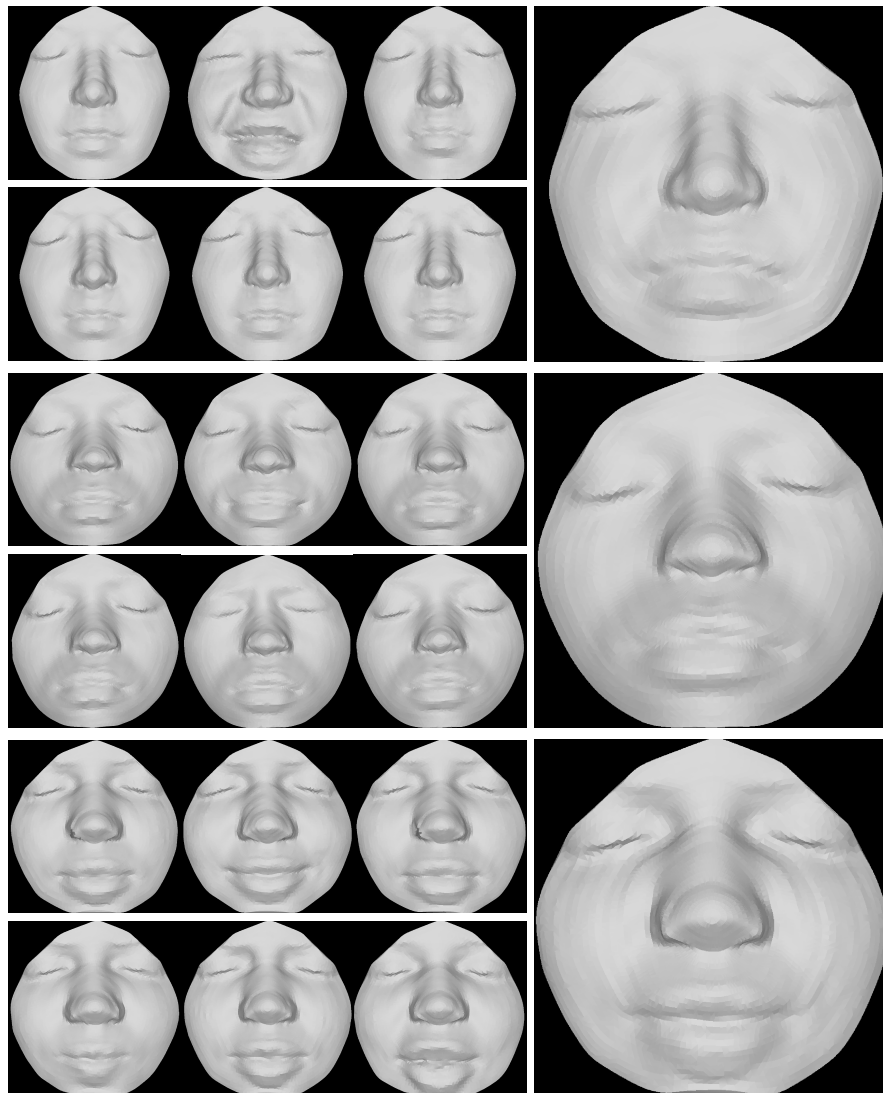


Fig. 13 The same person, at each row, under six facial expressions in the left panel, and the corresponding Karcher mean in the right panel.

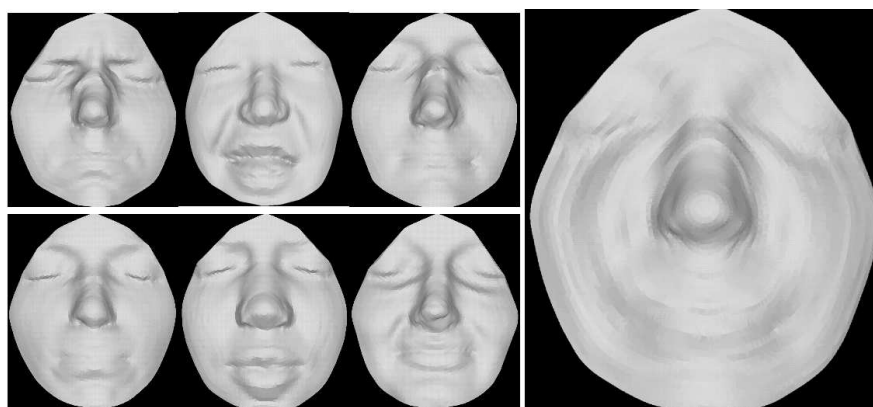


Fig. 14 Karcher mean (far right) for the six different people in the left panel.

6 Summary

In this paper we introduce a new computational framework for analyzing shapes of facial surfaces. The basic idea is to impose a specific coordinate system on facial surfaces using level curves of the surface distance function (measured from the tip of nose). Then, each surface can be represented as a path on the space of closed curves in \mathbb{R}^3 . We focus on the set of such paths and impose a Riemannian metric on it for defining geodesic paths. The main tool presented in this paper is the construction of geodesic paths between arbitrary two facial surfaces in the aforementioned set. There are multiple uses of this geodesic construction. The length of a geodesic between any two facial surfaces quantifies differences in their shapes; it also provides an optimal deformation from one to the other. Using Riemannian structure one can define simple statistics such as the sample mean, as is demonstrated for facial surfaces in this paper. A future application of this framework is in biometrics, for example in recognition of humans using shapes of their facial surfaces.

The central idea here is to decompose a surface into an indexed collection of closed curves and to compute geodesics between the corresponding curves on two faces. This geodesic is computed using the idea of path-straightening present in [9] which allows only the bending (and not stretching/compressing) of curves. There are other metrics and methods in the literature for computing geodesics between closed curves. An important idea is the use of elastic curves where the curves are allowed to stretch and compress for computing geodesic paths [12]. In future we propose to apply the framework of elastic curves to analyze facial curves; it requires extending the notion of elastic curves to curves in \mathbb{R}^3 .

Appendices

Let \mathcal{C} be a Riemannian manifold. Denote by I the unit interval $[0, 1]$. Denote by \mathcal{F} the space of measurable functions $[0, L] \rightarrow \mathcal{C}$. \mathcal{F} is a manifold; its tangent space is understood as follows. If $\alpha \in \mathcal{F}$, then

$$T_\alpha \mathcal{F} = \{w : [0, L] \rightarrow T\mathcal{C} : \forall \lambda \in [0, L], w(\lambda) \in T_{\alpha(\lambda)}\mathcal{C}\}$$

Clearly, this is just the set of first-order deformations of $\alpha \in \mathcal{F}$. We now make \mathcal{F} into a Riemannian manifold. If $w_1, w_2 \in T_\alpha \mathcal{F}$, define

$$\langle w_1, w_2 \rangle = \int_0^L \langle w_1(\lambda), w_2(\lambda) \rangle d\lambda$$

Theorem 1 *Suppose we are given a path in \mathcal{F} represented as $\phi : [0, L] \times I \rightarrow \mathcal{C}$. For each $\lambda \in [0, L]$, define $\phi_\lambda : I \rightarrow \mathcal{C}$ by $\phi_\lambda(t) = \phi(\lambda, t)$. Then ϕ is a geodesic in \mathcal{F} if $\forall \lambda \in [0, L]$, ϕ_λ is a geodesic in \mathcal{C} .*

Proof:

First, we review a useful characterization of geodesics in \mathcal{C} . Suppose v_0 and v_1 are two points in \mathcal{C} . Let \mathcal{B} denote the space of all paths in \mathcal{C} , parameterized by I , which start at v_0 and end at v_1 :

$$\mathcal{B} = \{\gamma : I \rightarrow \mathcal{C} \mid \gamma(0) = v_0 \text{ and } \gamma(1) = v_1\}.$$

Define an energy functional $E : \mathcal{B} \rightarrow \mathbb{R}$ by

$$E(\gamma) \equiv \frac{1}{2} \int_0^1 \langle \gamma'(t), \gamma'(t) \rangle dt,$$

Then, γ is a geodesic in \mathcal{C} , connecting v_0 and v_1 , if and only if the gradient of E with respect to γ is zero. In other words, γ is a critical point of the energy function E [17].

Now suppose that $\tilde{\phi} : [0, L] \times I \times (-\epsilon, \epsilon) \rightarrow \mathcal{C}$ is an arbitrary variation of ϕ in the space of curves in \mathcal{F} , i.e., we are assuming that for all $\lambda \in [0, L]$ and $t \in I$, $\tilde{\phi}(\lambda, t, 0) = \phi(\lambda, t)$ and for all $\lambda \in [0, L]$ and $h \in (-\epsilon, \epsilon)$, $\tilde{\phi}(\lambda, 0, h) = \phi(\lambda, 0)$ and $\tilde{\phi}(\lambda, 1, h) = \phi(\lambda, 1)$. For each value of $h \in (-\epsilon, \epsilon)$, we calculate the energy of the path $\tilde{\phi}(\cdot, \cdot, h)$ in \mathcal{F} as follows:

$$\begin{aligned} E(\tilde{\phi}(\cdot, \cdot, h)) &= \frac{1}{2} \int_0^1 \int_0^L \left\langle \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h), \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h) \right\rangle d\lambda dt \\ &= \frac{1}{2} \int_0^L \int_0^1 \left\langle \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h), \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h) \right\rangle dt d\lambda \end{aligned}$$

Differentiating with respect to h at $h = 0$ gives:

$$\frac{d}{dh}_{h=0} E(\tilde{\phi}(\cdot, \cdot, h)) = \frac{1}{2} \int_0^L \frac{d}{dh}_{h=0} \left(\int_0^1 \left\langle \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h), \frac{\partial \tilde{\phi}}{\partial t}(\lambda, t, h) \right\rangle dt \right) d\lambda$$

If we assume that ϕ_λ is a geodesic in C for every $\lambda \in [0, L]$, then it follows immediately that the function we are integrating over $[0, L]$ in the right hand side of the above expression is 0 for every λ , proving that ϕ is a geodesic in \mathcal{F} .

References

1. A. M. Bronstein, M.M. Bronstein, and R. Kimmel. Three-dimensional face recognition. *International Journal of Computer Vision*, 64(1):5–30, 2005.
2. K. I. Chang, K. W. Bowyer, and P. J. Flynn. An evaluation of multimodal 2D+3D face biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):619–624, 2005.
3. K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE Pattern Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, 2006.
4. E. W. Dijkstra. *A note on two problems in connection with graphs*. Numerische Mathematik, 1959.
5. J. Glaunes, A. Trounev, and L. Younes. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *CVPR (2)*, pages 712–718, 2004.
6. P.W. Hallinan, G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Three-Dimensional Patterns of Face*. A. K. Peters, 1999.
7. M. Hilaga, S. T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *ACM SIGGRAPH*, Annual Conference Series, page 203212, 2001.
8. N. Khaneja, M. I. Miller, and U. Grenander. Dynamic programming generation of curves on brain surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1260–1265, 1998.
9. E. Klassen and A. Srivastava. Geodesic between 3D closed curves using path straightening. In *European Conference on Computer Vision (ECCV)*, page ??, 2006.
10. E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Patt. Analysis and Machine Intell.*, 26(3):372–383, March, 2004.
11. P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *Journal of the European Mathematical Society*, 8:1–48, 2006.
12. W. Mio, A. Srivastava, and S. Joshi. On shape of plane elastic curves. *International Journal of Computer Vision*, 73(3):307 – 324, July 2007.
13. R. Osada, T. Funkhouser, B. Chazells, and D. Dobkin. Matching 3D models with shape distributions. In *IEEE Shape Modeling International*, May 2001.
14. C. Samir, A. Srivastava, and M. Daoudi. Three-dimensional face recognition using shapes of facial curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1847–1857, 2006.

15. T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.
16. J. Shah. An H^2 type riemannian metric on the space of planar curves. In *Workshop on the Mathematical Foundations of Computational Anatomy, MICCAI2006*, October, 2006.
17. Michael Spivak. *A Comprehensive Introduction to Differential Geometry, Vol I & II*. Publish or Perish, Inc., Berkeley, 1979.
18. V. Surazhsky, T. Surazhsky, D. Kirsanov, S. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. In *ACM SIGGRAPH*, pages 553–560, 2005.
19. S. Wang, Y. Wang, M. Jin, Xianfeng Gu, and Dimitris Samaras. 3D surface matching and recognition using conformal geometry. In *CVPR (2)*, pages 2453–2460, 2006.
20. S. Yoshizawa, A. G. Belyaev, and H.-P. Seidel. Fast and robust detection of crest lines on meshes. In *Proc. of ACM Symposium on Solid and Physical Modeling*, pages 227–232, 2005.
21. L. Younes. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58:565–586, 1998.