Noname manuscript No.
(will be inserted by the editor)

An Intrinsic Framework for Analysis of Facial Surfaces

Chafik Samir[1], Anuj Srivastava[2], Mohamed Daoudi[1], Eric Klassen[3]

[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 statistical analysis of shapes of facial surfaces can play an important role in biometric authentication and other face-related applications. The main difficulty in developing such an analysis comes from the lack of a canonical system to represent and compare all facial surfaces. This paper suggests a specific, yet natural, coordinate system on facial surfaces, that enables comparisons of their shapes. Here a facial surface is represented as an indexed collection of closed curves, called facial curves, that are level curves of a surface distance function from the tip of the nose. Defining the space of all such representations of face, this paper studies its differential geometry and endows it with a Riemannian metric. It presents numerical techniques for computing geodesic paths between facial surfaces in that space. This Riemannian framework is then used to: (i) compute distances between faces to quantify differences in their shapes, (ii) find optimal deformations between faces, and (iii) define and compute average of a given set of faces. Experimental results generated using laser-scanned faces are presented to demonstrate 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 such 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 medical diagnostics, object designs, database search, and some forms of biometrics. Focusing on the last topic, our goal in this paper is to develop a mathematical and statistical 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*. In Kendall’s definition of shape [11], shape is a property that is invariant to rigid translations and rotations, and uniform scalings of the objects. In our analysis of facial shapes, however, we have chosen the set of shape-preserving transformations to be translations and rotations only; scaling a facial surface will change its shape in our framework. Emphasizing specific goals, we want to develop a mathematical framework for representing and comparing shapes of facial surfaces, using metrics that are invariant to

3D rotations and translations. In this process, we will also obtain optimal deformations of facial surfaces from one into another. Our representations are chosen such that the ensuing face comparisons are stable with respect to deformations of facial surfaces that results from different facial expressions. This is an important advantage of this framework, 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 help understand this issue, consider the problem of analyzing shapes of curves. In this case, there is a fixed ordering of points along the curves that allows their parameterizations. This idea allows the use of parametric representations of curves for analyzing their shapes in a manner that is invariant to the choice of parametrization [23,16,14,17,24,34]. 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. So, the registration (or correspondence) of points across surfaces is a more difficult problem. 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 a surface into a collection of curves and the correspondence across surfaces is obtained at two levels: first between curves and then between points along corresponding curves.

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 [4,3,2]. 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 [18] and conformal geometry [32], 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 of the few exceptions include Glaunes et al. [7,31] 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 unlabelled points-sets. Another approach that is used to analyze surfaces in general is to use a spectral analysis of the Laplacian-Beltrami operator on them. For instance, Rustamov [20] uses a "global point signature", defined using the eigen values and eigen functions of the Laplace-Beltrami operator, to perform matching and classification of surfaces. A similar spectral representation has been used by Jain et al. [10] to perform matching of 3D shapes. Although these spectral methods are successful in modeling surface geometry in an intrinsic fashion, invariant to rigid motions, they do not account for different re-parameterizations of surfaces. There is also a significant literature on analyzing shapes of 3D objects using Morse functions and Reeb graphs [6,25,30]. We believe that while this framework is important for analyzing shape of general 3D objects, with widely varying geometries and even topologies, this may not be as effective in comparing shapes of surfaces, such as faces, with identical topologies and similar geometries.
- **Shape Analysis of Curves**: There has been a significant amount of research in analyzing shapes of open and closed curves, especially in a plane. The common theme has been to derive spaces of curves, with different representations of curves, and to study the differential geometries of these spaces modulo the space of re-parameterizations. In the

case of closed curves, one uses an additional constraint to ensure closure; this constraint results in the set of relevant curves being a nonlinear sub-manifold inside some Hilbert space. To compare any two curves, one inherits a Riemannian structure from the Hilbert space and computes geodesic paths on the sub-manifold.

- **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 [12]. 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 [8], or use surface gradients to extract crest lines[29,33]. However, computations of gradients and curvatures both involve higher order derivatives and, as mentioned above, are numerically unstable. In [21,22], Samir et al. suggested the use of level sets of the height function (z coordinates) as curves of interest for face analysis. 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. A solution is given by Bronstein et al. [1] who use a surface distance function to define level curves that are invariant to rigid motions. Secondly, past works have been intended for face recognition only and not for a Riemannian analysis of facial surfaces. In other words, they lack a formal representation of facial surfaces as elements of a Riemannian manifold that can lead to formal geodesic paths (or optimal deformations) between faces. Consequently, interesting statistical quantities such as the mean face were neither defined nor computed.

1.2 Our Approach

Our approach is to represent facial surfaces as indexed collections 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 [21] 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 [21], but the definition of facial curves is now different. Facial curves are level curves of a *surface distance function* 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. Bronstein et al. [1] have argued that this function is stable with respect to changes in facial expressions and have used it to crop facial surfaces at their boundaries. Hilaga [9] and others (see e.g. [30]) have used it to construct Reeb graphs of more general 3D objects. An important property of this function is that it is invariant to rotation and translation of the facial surface. Visually, the level curves of this function seem to contain distinct features that can prove to be important in face classification and recognition. (These curves are closed curves in $\mathbb{R}^3$ while the curves used in [21] were 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 elements of this space. This tool for computing geodesic is shown to be useful in: (i) finding optimal deformations between faces, (ii) for registering points of facial surfaces, and (iii) for computing "average faces".

The rest of this paper is organized as follows. A mathematical representation of a facial surface, using facial curves, is introduced in Section 2, along with a step-by-step procedure

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

Symbol	Definition /Explanation
S	a smooth facial surface without holes
$dist$	length of the shortest path on S between any two points
λ	a variable for the value of $dist$ from the tip of the nose
$c_\lambda(s)$	level curve of $dist$ on S at the level λ, parameterized by s such that $\frac{dc_\lambda(s)}{ds}$ =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, $v_\lambda(s) = \frac{1}{l_\lambda}\frac{dc_\lambda(s)}{ds}$
v_0	the velocity function of a unit circle in the XY plane, $v_0(s) = [-\sin(s) \ \ \cos(s) \ \ 0]'$
(a, b)	the Euclidean inner product in $\mathbb{R}^3$
$\mathcal{C}$	the set of all closed curves in $\mathbb{R}^3$,
	$\mathcal{C} = \{v \mid v : [0, 2\pi] \to \mathbb{S}^2,\ \int_0^{2\pi} v(s)ds = 0,\ (v(s), v(s)) = 1, \forall s\}$
$T_v\mathcal{C}$	the space of all tangents to $\mathcal{C}$ at v
	it is given by $\{f \mid f : [0, 2\pi] \to \mathbb{R}^3,\ \forall s,\ (f(s), v(s)) = 0\ \}$
$\langle f_1, f_2\rangle$	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 $t \in [0, 1]$
	$\phi_0(v_1, v_2) = v_1,\ \phi_1(v_1, v_2) = v_2$
α	an indexed collection of closed curves in $\mathbb{R}^3$
	$\alpha : [0, L] \to \mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+,\ \alpha(\lambda) = (p_\lambda, v_\lambda, l_\lambda)$
$\mathcal{H}$	the space of all parameterized paths in $\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+$,
	$\mathcal{H} = \{\alpha : [0, L] \to (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+), \alpha(0) = (r, v_0, 0)\}$
$T_\alpha\mathcal{H}$	the tangent space of $\mathcal{H}$ at α,
	$T_\alpha\mathcal{H} = \{(u, w, x) \mid u(\lambda) \in \mathbb{R}^3, w(\lambda) \in T_{\alpha(\lambda)}\mathcal{C}, x(\lambda) \in \mathbb{R}\}$
$\langle\langle(u_1, w_1, x_1), (u_2, w_2, x_2)\rangle\rangle$	the Riemannian metric on $\mathcal{H}$
	$= \int_0^L ((u_1(\lambda), u_2(\lambda)) + \langle w_1(\lambda), w_2(\lambda)\rangle + x_1(\lambda)x_2(\lambda))\, d\lambda$
$\psi_t(\alpha_1, \alpha_2)$	the geodesic path in $\mathcal{H}$ from α_1 to α_2 parameterized by t

to generate this representation. A Riemannian analysis of closed curves in $\mathbb{R}^3$ is presented in Section 3, with its extension to Riemannian analysis of facial surfaces outlined in Section 4. The notion of Karcher mean is applied to facial surfaces in Section 5 and the paper finishes with a brief summary in Section 6.

2 Mathematical Representation of a Facial Surface

Consider a facial surface S to be a two-dimensional smooth, connected manifold with 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 will start by assuming that it is a continuous surface. Some pictorial examples of S are shown in Figure 1 where 2D images and their corresponding facial surfaces for 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 closed curves or, equivalently, as a parameterized path in the space of closed curves, that allows the decomposition of a 2D registration problem into two independent 1D registration problems. Let r be a prominent reference point on S that is easily detectable. In this paper, we have chosen $r \in S$ as the tip of the nose. Then, we define a function $dist : S \times S \to \mathbb{R}$ to be the **surface distance function** from r to

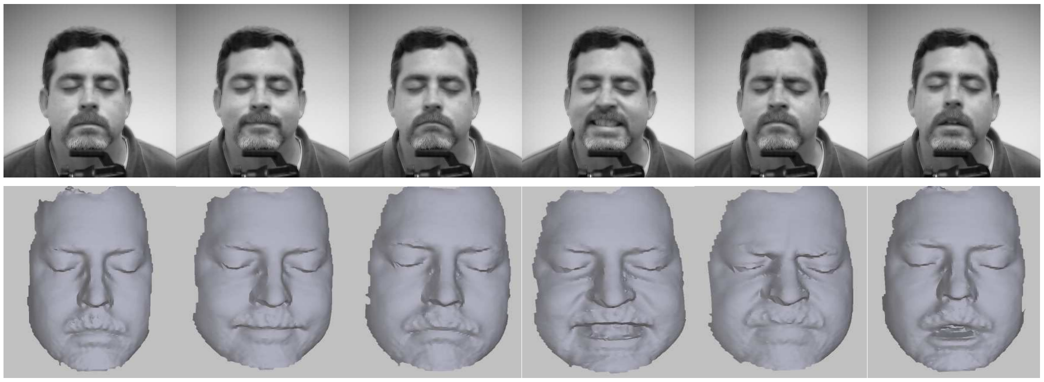


Fig. 1 2D images and their corresponding facial surfaces of same person under different facial expressions: neutral, smile, frown, angry, squint, and scared.

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 the function $dist(r, \cdot)$:

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

For large values of λ, c_λ is naturally empty while for $\lambda = 0$, c_λ is the singleton $\{r\}$. For values of λ in between those two extremes, we expect c_λ to capture certain local geometry of S. So far c_λ is just a collection of points but these points can be ordered to form a parameterized, closed curve $c_\lambda(s)$ that is parameterized by constant speed over the interval $[0, 2\pi]$, i.e. $\frac{dc_\lambda(s)}{ds} =$ 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.
- **Velocity function**: Define the velocity function $v_\lambda : [0, 2\pi] \to \mathbb{S}^2$, where $v_\lambda(s) = \frac{1}{\|\frac{dc_\lambda(s)}{ds}\|} \frac{dc_\lambda(s)}{ds}$.

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 XY 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 points towards the boundary of S.

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

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

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

2.1 Mesh Pre-Processing

In practice one does not have continuous surfaces but rather discrete meshes, obtained by sampling facial surfaces in an irregular fashion. The extraction of facial curves from such

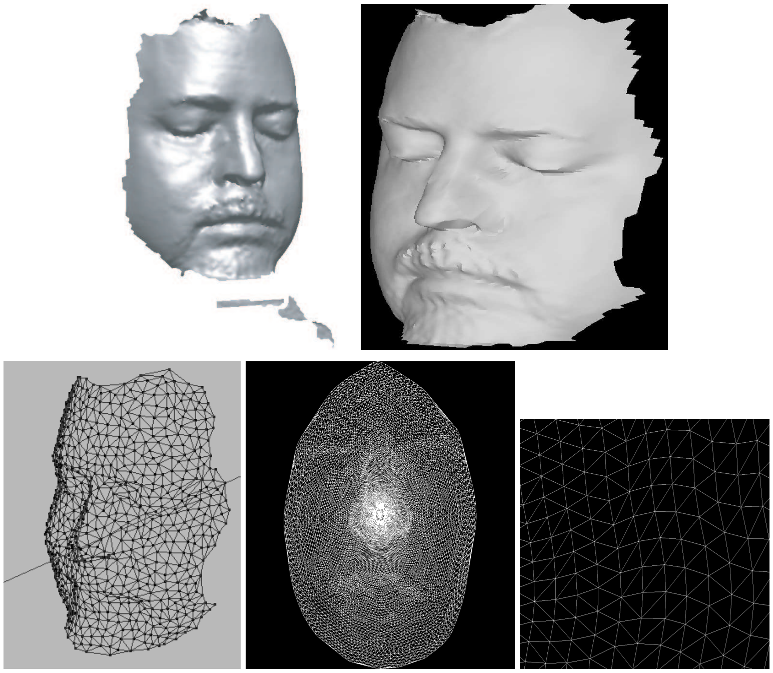


Fig. 2 Top: the scanned surface (left) and the surface after removal of disconnected parts (right). Bottom: A nonuniform mesh before regularization (left), the uniform mesh after regularization (middle), and a part of regularized mesh enlarged (right).

meshes involves several steps to ensure robustness. First we remove disconnected parts so that the surface distance function *dist* is well defined. Shown in the top row of Figure 2 is an example of this removal. The next step is to fill holes in the mesh. Since holes denote missing points from the mesh, we need to use the observed parts to predict the missing parts. We use a uniform sampling and a linear interpolation between visible points to fill in the missing points. Once we have a complete mesh, i.e. without holes, we regularize them using routines from the GNU triangulated surface library. This step forces the triangles in the mesh to be within a certain size interval – larger triangles are made small by division and smaller triangles are made large by union. This step is not energy driven; it is simply based on a sorted list of triangles according to their edge lengths and treating them in that order. Finally, using the same library, we can increase the number of vertices to obtain a finer resolution. Shown in the bottom row of Figure 2 is an illustration of preprocessing a facial surface. The original, non-uniform mesh is shown in the left panel and the regularized mesh is shown in the middle panel. The last panel shows a zoom of a part of the regularized mesh to demonstrate its uniformity.

2.2 Surface Distance Function dist

Once a facial surface is pre-processed, the tip of the nose can be detected using standard methods. We use a combination of the location and the curvature to find the tip. Although there is a possibility of variability in this detection step, due to mesh noise and other measurement errors, we have found that the regularization and smoothing steps in the pre-processing help stabilize this part. Several methods have been proposed in the past to compute distances on discrete graphs or meshes. Considering the trade off between compu-

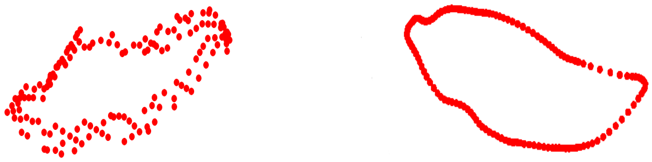


Fig. 3 Level set extraction. Left: c_λ for a larger δ and for un-regularized mesh. Right: c_λ for a small δ and for a regularized and refined mesh.

tational cost and accuracy, we employ a relatively simple method in which geodesic distance is approximated by Dijkstra's algorithm [5] based on the edge lengths. It has a computational cost of order $O(n \log(n))$ with n being the number of vertices in the mesh. Other techniques, such as those presented in [28] or [15], can also be used instead.

2.3 Facial Curves Extraction

Once we have a tool for computing *dist* on the mesh, we consider the task of extracting a level curve c_λ. In practice, the definition of c_λ is expanded to include a (thin) strip rather than a curve. Now c_λ consists of all points whose distance *dist* from r is in $[\lambda - \delta, \lambda + \delta]$, for a small $\delta > 0$. Once *dist* has been computed for all vertices in the mesh, the extraction of c_λ is simply a selection of appropriate vertices. The selected points are then ordered using the Euclidean Minimum Spanning Tree algorithm to approximate a parameterized curve. To this end, we first form a graph that consists of the given set of points as its nodes and the pairwise Euclidean distances between them as the edge weights. Then, we search for the optimal spanning tree of this graph, and apply an averaging technique iteratively to smooth the curve. To specify the starting point p_λ for a given curve c_λ, in this work, we choose p_λ to be the intersection between c_λ and the plane containing the bridge of the nose.

Shown in Figure 3 are examples of extracted c_λ corresponding to two different values of δ: large (left panel) and small (right panel). To further emphasize the role of regularization, the left set is for an un-regularized mesh while the right panel is for its regularized version. Note that δ is proportional to the distance between two nodes on the regularized mesh, and when δ is small enough, the inter-set distances from c_λ are relatively large. Using this fact, the ordering and smoothing process is very successful, and doesn't significantly affect the shape of the curve. Figure 4 shows more examples of extracted facial curves, drawn directly on the original surface. The number of curves increases from left to right.

Returning to the issue of errors in detection of the reference point (nose tip), an important question is: How are the shapes of facial curves affected by small changes in the location of the reference point? In our experiments, these shapes are found to be quite stable to small variations in reference location. This is due to the regularization of facial surfaces in the pre-processing.

We summarize the steps needed to extract c_λ from a discrete observation of S in Algorithm 1.

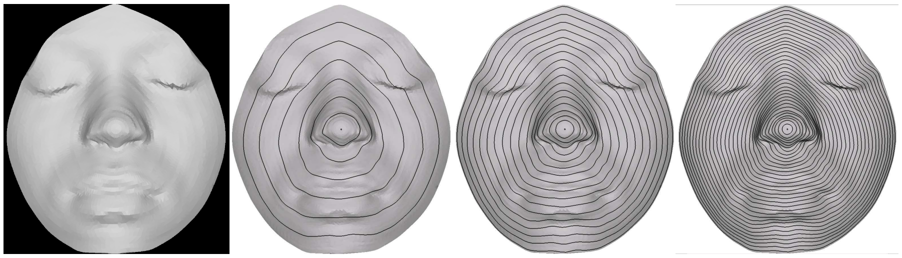


Fig. 4 Representation of a facial surface by an indexed collection of curves. For the surface shown in the left panel, we show an increasing number of level curves of $dist(r, \cdot)$ from left to right.

Algorithm 1 : Facial curve extraction and parametrization

```
Given a facial surface $S$, the $c_\lambda$s are determined as follows:
  1- Preprocess the surface according to Section 2.1.
  2- Calculate dist value for 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 $c_\lambda$.
  6- Uniformly re-sample $c_\lambda$.
  7- Compute the tangent vector function $v_\lambda(s)$ and the curve length $l_\lambda$; and set the parametrization $c_\lambda = (p_\lambda, v_\lambda, l_\lambda)$.
```

2.4 Face Representation Using Collection of Facial Curves

Now we are ready to state our mathematical representation of a facial surface. Consider a path $\alpha : [0, L] \to (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+)$, 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.

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

We will represent facial surfaces and analyze them as elements of $\mathcal{H}$. Note that not all elements of $\mathcal{H}$ are reasonable representations of facial surfaces. This issue is complicated by the lack of a formal definition of what is a facial surface. Our idea is to represent faces as elements of this larger set $\mathcal{H}$ and to perform analysis in $\mathcal{H}$. Shown in Figure 5 are examples of facial surfaces represented by their corresponding elements in $\mathcal{H}$. The top row shows a number of surfaces S while the bottom row shows the corresponding elements of $\mathcal{H}$. 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 shall ultimately depend on the application.

We note the following properties of this representations:

1. **Invariance to Rotation and Translation**: The choice of the surface distance function *dist* to define facial curves ensures the invariance shapes of level curves c_λ to rotation and translation of S. Since α is simply a collection of facial curves c_λs, 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_λs 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, S and α are identical as

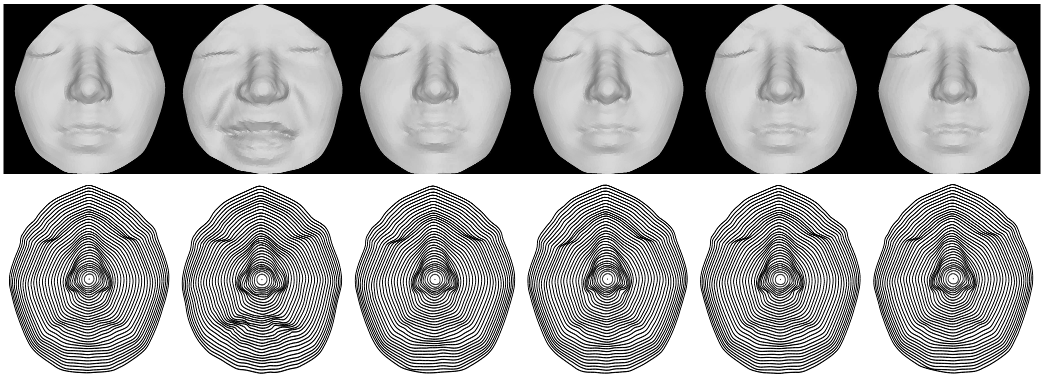


Fig. 5 Top row: Facial surfaces of the same person under different facial expressions. Bottom row: Their correspondents representations in $\mathcal{H}$.

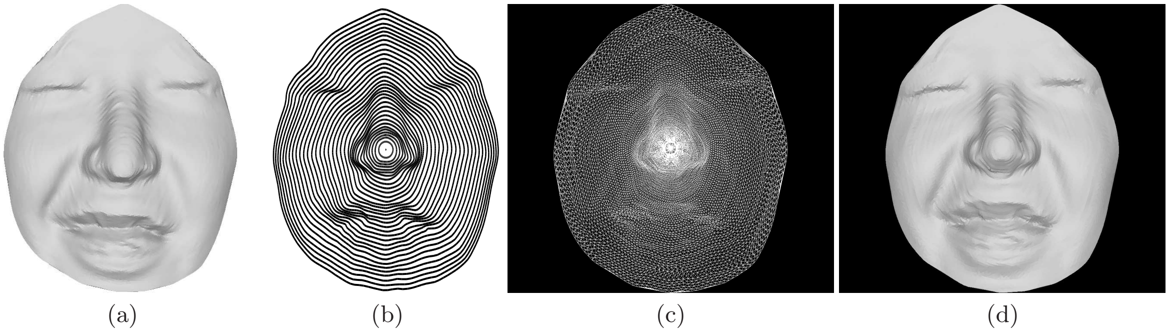


Fig. 6 Reconstruction example: (a) an original facial surface, (b) its representation as $\alpha \in \mathcal{H}$, as a collection of curves, (c) a Delauney triangulated mesh formed from α, and (d) the reconstructed surface.

sets of points in $\mathbb{R}^3$. Can we reconstruct a facial surface using its representation $\alpha \in \mathcal{H}$? Yes. We can generate a Delaunay triangulation of the set of points contained in α and use this triangulation to render a surface. An example is shown in Figure 6 which shows the original surface (left), a collection of facial curves (second), the corresponding dense mesh generated using Delaunay triangulation (third), and the rendered facial surface using that triangulation (right).

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.

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 p and l 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. [13], and we will adapt that solution to our case. 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 to understand its tangent bundle and to impose a Riemannian structure on it. Recall that for any $v \in \mathcal{C}$, we have $\|v(s)\| = 1$; thus, v can also be seen as a path on $\mathbb{S}^2$ and a vector f tangent to $\mathcal{C}$ at v becomes a set of vectors tangent to $\mathbb{S}^2$ along the path v. The space of all such tangent vectors, denoted by $T_v(\mathcal{C})$, is given by:

$$T_v(\mathcal{C}) = \{f | f : [0, 2\pi] \to \mathbb{R}^3, \forall s \ \ (f(s) \cdot v(s)) = 0, \ \int_0^{2\pi} f(s)ds = 0 \ \} \ . \tag{3}$$

The last condition comes from the fact that $\int_0^{2\pi} v(s)ds = 0$ for a closed curve. To impose a Riemannian structure on $\mathcal{C}$, we will assume the following inner product on $T_v(\mathcal{C})$: for $f, g \in T_v(\mathcal{C})$,

$$\langle f, g \rangle = \int_0^{2\pi} (f(s) \cdot g(s))ds \ . \tag{4}$$

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 [13]. The basic idea is to start with any path $\phi(t)$ connecting v_0 and v_1. That is $\phi : [0, 1] \to \mathcal{C}$ 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:

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

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_c until the gradient becomes zero. It has been shown in [26] (in general) and in [13] for our case, that a critical point of E_c is a geodesic on $\mathcal{C}$. Shown in Figure 7 are two examples of geodesic paths between two facial curves and the corresponding evolution of E_c during path straightening.

Remark: One difference between the framework in [13] and here is that the origin on the individual facial curves are kept fixed here. In [13], the variability due to origins 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 and the tip 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 $\mathcal{C}/\mathbb{S}^1$. The reason for doing so is to reduce the computational cost.

4 Riemannian Analysis of Faces as elements of $\mathcal{H}$

Now 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 depends upon a Riemannian metric, which in this case measures amount of work done in bending of facial surfaces. To actually compute geodesics, we need to study the differential geometry of $\mathcal{H}$. The tangent space of $\mathcal{H}$ is given by:

$$T_\alpha(\mathcal{H}) = \{(u, w, x) | \ \forall \lambda \in [0, L], \ \ u_\lambda \in \mathbb{R}^3, \ w_\lambda \in T_{v_\lambda}(\mathcal{C}), \ \ x_\lambda \in \mathbb{R}\} \ , \tag{5}$$

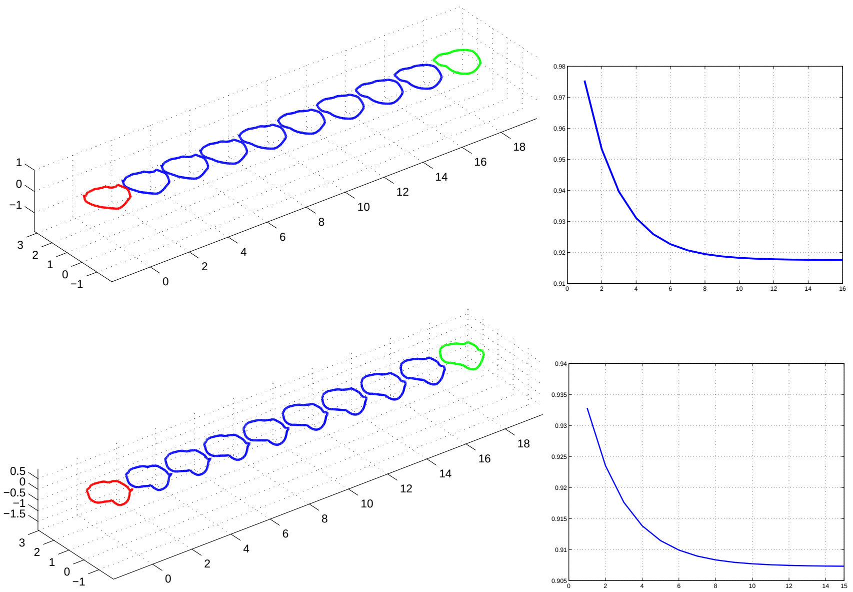


Fig. 7 Two examples of geodesic paths between facial curves. Each row shows a geodesic path between the end curves (left) and the evolution of energy E_c in finding this geodesic (right).

where $T_{v_\lambda}(\mathcal{C})$ is as specified in Eqn. 3. $\mathcal{H}$ is an infinite-dimensional nonlinear manifold and we give it a Riemannian structure by imposing the metric:

$$\langle\langle (u_1, w_1, x_1), (u_2, w_2, x_2) \rangle\rangle = \int_0^L ((u_1(\lambda), u_2(\lambda)) + \langle w_1(\lambda), w_2(\lambda)\rangle + x_1(\lambda) x_2(\lambda))\, d\lambda \ , \quad (6)$$

where $(\cdot,\cdot)$ is the Euclidean inner product in $\mathbb{R}^3$ and $\langle\cdot,\cdot\rangle$ 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 $\alpha_i(\lambda) = (p_{i,\lambda}, v_{i,\lambda}, l_{i,\lambda}) \in (\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+)$. For each of the components we compute the geodesic paths independently by defining: for $t \in [0, 1]$

$$\Psi(t) = \{\Psi_\lambda(t), \lambda \in [0, L]\}, \ \text{ where } \ \Psi_\lambda(t) = (\Psi^p_\lambda(t), \Psi^v_\lambda(t), \Psi^l_\lambda(t)) \ . \quad (7)$$

Ψ_λ is the geodesic path between the corresponding facial curves; it is composed of three components. Ψ^p_λ is the geodesic for the position component, Ψ^v_λ is for the shape component, and Ψ^l_λ is for the length component. These individual components are computed as follows:

1. **Shape of Curves**: Let $\Psi^v_\lambda(t)$ be a geodesic path in $\mathcal{C}$, constructed using path straightening as discussed in Section 3, such that $\Psi^v_\lambda(0) = v_{0,\lambda}$ and $\Psi^v_\lambda(1) = v_{1,\lambda}$.
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, $\Psi^p_\lambda : [0, 1] \to \mathbb{R}^3$, $t \to tp_{1,\lambda} + (1-t)p_{0,\lambda}$.
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, $\Psi^l_\lambda : [0, 1] \to \mathbb{R}^+, t \to tl_{1,\lambda} + (1-t)l_{0,\lambda}$.

Proposition 1 *The path Ψ defined in Eqn. 7 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 $(\Psi^p_\lambda, \Psi^v_\lambda, \Psi^l_\lambda)$ are individually geodesics in the 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. 6. As shown in Appendix, the set of all geodesics $\Psi = \{\Psi_\lambda | \lambda \in [0, L]\}$ is a geodesic in $\mathcal{H}$ if each one of Ψ_λ is a geodesic in $(\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+)$. □

We have summarized the main steps for computing geodesic paths between given two facial surfaces in $\mathcal{H}$ in Algorithm 2.

Algorithm 2 : Computation of geodesic between S_1 and S_2

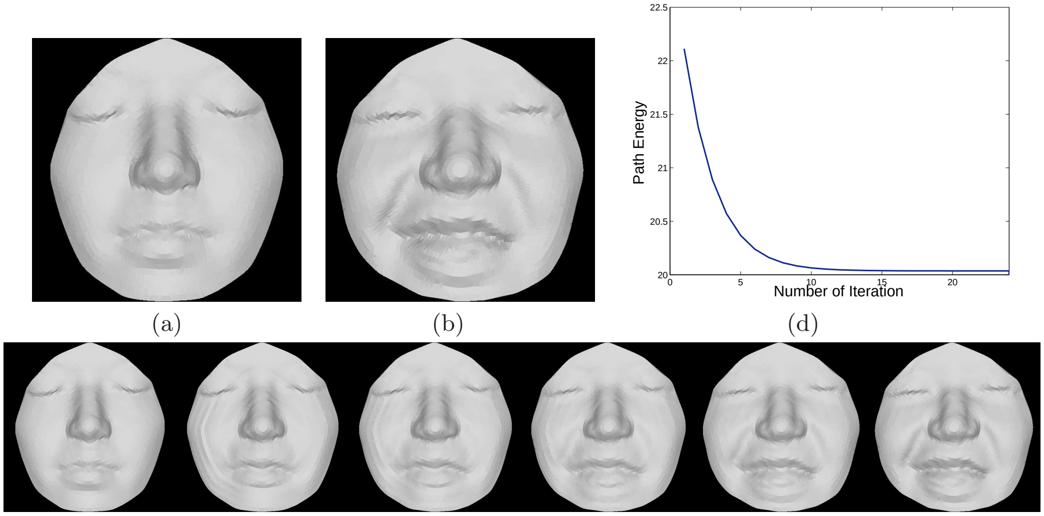
Given two different surfaces S_1 and S_2, this algorithm summarizes computational steps to construct a geodesic path between their representations in $\mathcal{H}$.

```
1- Extract level curves $c^1_\lambda$ and $c^2_\lambda$, $\lambda \in [0, L]$, for $S_1$ and $S_2$, respectively, using Algorithm 1.
2- Represent each surface as an element of $\mathcal{H}$ and denote them by $\alpha_0$ and $\alpha_1$.
3- Compute the geodesics $\Psi_\lambda(t)$, at each level $\lambda \in [0, L]$, in $\mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+$.
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 $\{\Psi_\lambda(t), \lambda \in [0, L]\}$.
```

We took facial surfaces of a number of people under different facial expressions and constructed geodesic paths between them. Some examples are shown in Figures 8 - 11. Figures 8(a) and (b) show two facial surfaces of the same person under the two expressions: neutral and smile, respectively and (c) shows the resulting geodesic path between them. Drawn in (c) are facial surfaces denoting five equally spaced points in $\mathcal{H}$ along the geodesic path $\Psi(t)$. With respect to the chosen Riemannian metric, this path denotes the optimal deformation from one face to another, and the path-length quantifies the amount of deformation. Two additional examples of geodesics are shown in Figure 9. In 9(a) we show a geodesic path between two facial surfaces of the same person, while in 9(b) we show the same for faces belonging to different persons. Each path has been shown from three different viewpoints to help illustrate the 3D deformation. Figures 10 and 11 show several additional examples of these geodesic paths from the front view.

In some cases it is important to study the deformations resulting from geodesic paths between facial surfaces. Since these deformations are optimal, in the sense of geodesic lengths, they can provide important visual information about variability of facial shapes. There are several ways to visualize these deformations; a simple way is to compute the magnitude of the initial velocity vector, $\frac{d}{dt}\Psi(t)|_{t=0}$, and display it as a scalar field on the initial surface $\Psi(0)$. This is presented in Figure 12 where deformation fields resulting from geodesic paths from one face to several others are shown. The top row is for different faces of the same person while the bottom row is for faces from different persons. It can be seen in the top row that the deformation field is larger when a person changes from neutral expression to smile, due to deformations near the lips and cheeks. For the remaining faces in the upper row, the expressions were less pronounced and, consequently, the deformations are rather small. The deformations across persons are relatively much larger, as can be seen from the examples in the bottom row.

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 13 are two examples of such experiments. Each of the 12 × 12 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 images in Figure 13; the brightness



(a)

(b)

(d)

(c)

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.

of a pixel is proportional to the geodesic distance between the corresponding faces. Since the last six faces belong to different people, in each case, these images correctly shows brighter rows and columns beyond the first 6 × 6 block.

5 Karcher Means of Facial Surfaces

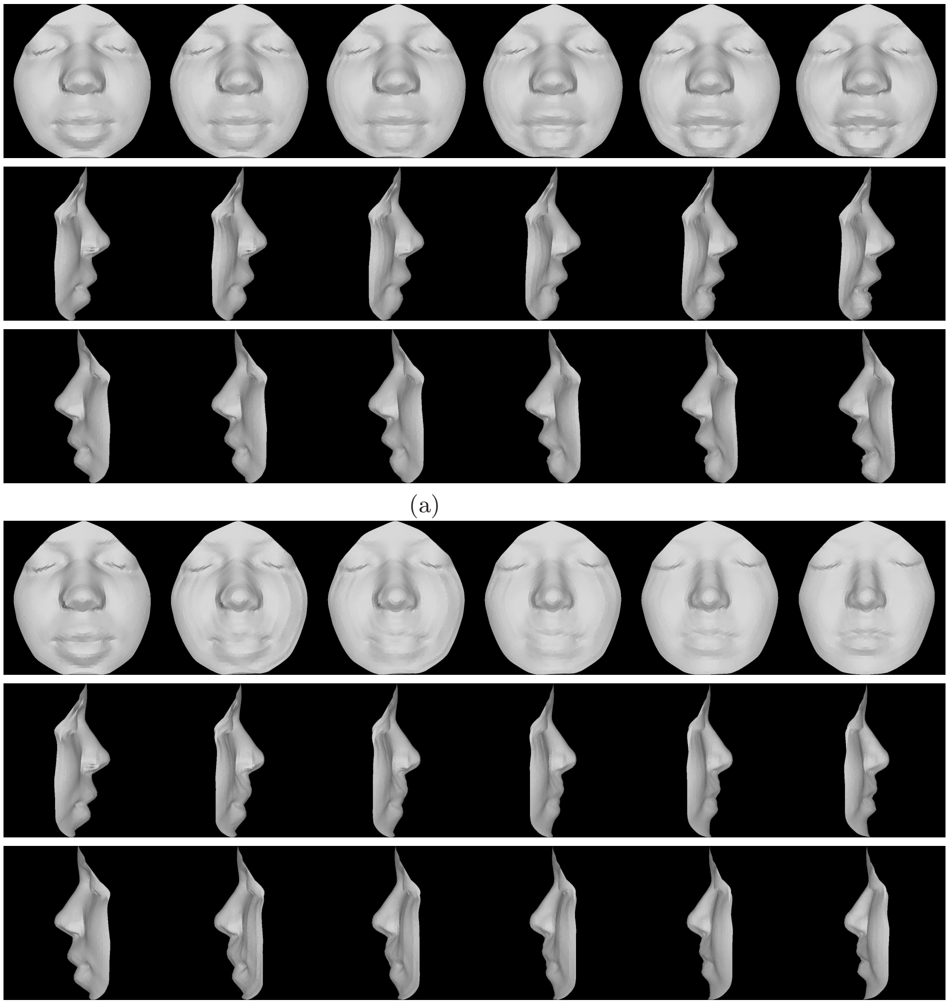
For future statistical analysis, we are interested in defining a notion of "mean" for a given set of facial surfaces. The Riemannian structure defined on $\mathcal{H}$ enables us to perform such statistical analysis for computing means and variances of faces. There are at least two ways of defining a mean value for a random variable that takes values on a nonlinear manifold. The first definition, called the *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 calculate the Karcher mean of facial surfaces $\{S_1, ..., S_n\}$ in $\mathcal{H}$, define the variance function:

$$\mathcal{V} : \mathcal{H} \to \mathbb{R}, \mathcal{V}(S) = \sum_{i=1}^{n} d(S_i, S_j)^2 \tag{8}$$

The Karcher mean is then defined by:

$$\overline{S} = \arg\min_{\mu \in \mathcal{H}} \mathcal{V}(\mu) \tag{9}$$

The intrinsic mean 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, $\overline{S}$ is an element of $\mathcal{H}$, that has the smallest total deformation from all given surfaces.



(a)

(b)

Fig. 9 Geodesic paths between: (a) faces from the same person under different facial expressions, viewed from three different viewpoints, and (b) faces from different persons.

We present a commonly used algorithm for finding Karcher mean for a given set of facial surfaces. This approach, presented in Algorithm 3, uses the gradient of $\mathcal{V}$, in the space $T_\mu(\mathcal{H})$, to update the current mean μ. Since this is a gradient approach, it only ensures a local minimizer of the variance function $\mathcal{V}$. The underlying manifold $\mathcal{H}$ seems too complicated to derive any additional conclusions about reaching a global minimizer of $\mathcal{V}$. For a discussion on uniqueness and existence of intrinsic means, please refer to Pennec [19].

Several examples of using the Karcher mean to compute average faces are 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 different persons. Note that the facial surfaces of the same person, under different facial expressions are quite close to their Karcher mean. For a clustering or an identification purposes, they could be reasonably represented by their Karcher mean. However, the Karcher mean of faces of different people

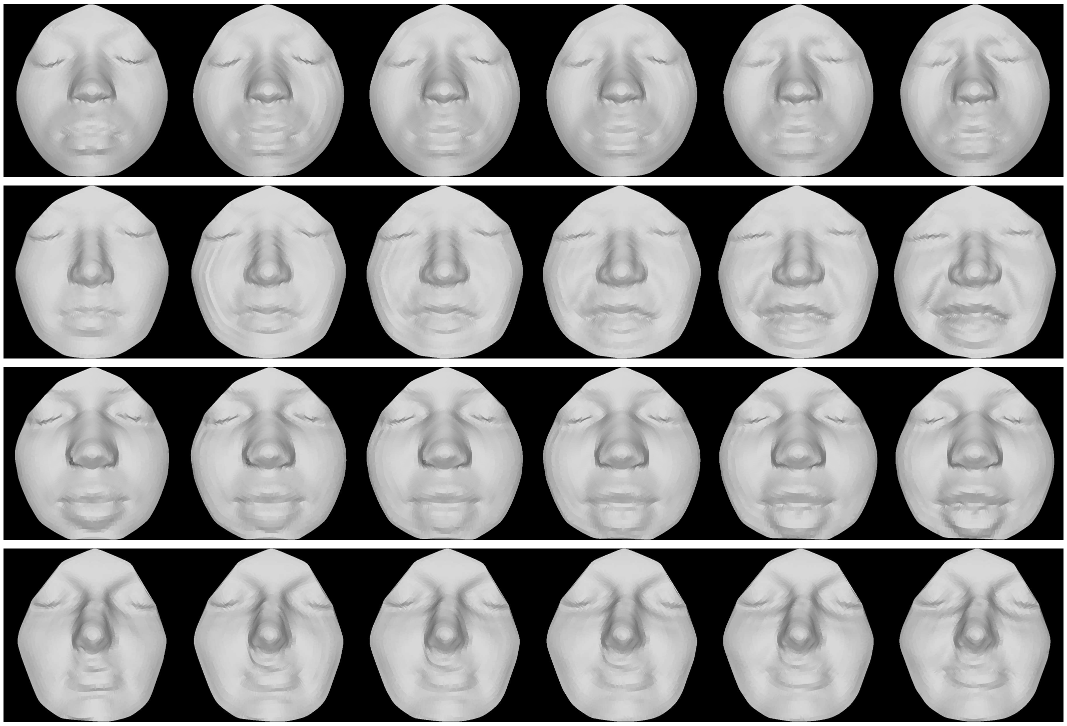


Fig. 10 Geodesic paths between facial surfaces of same person under different expressions.

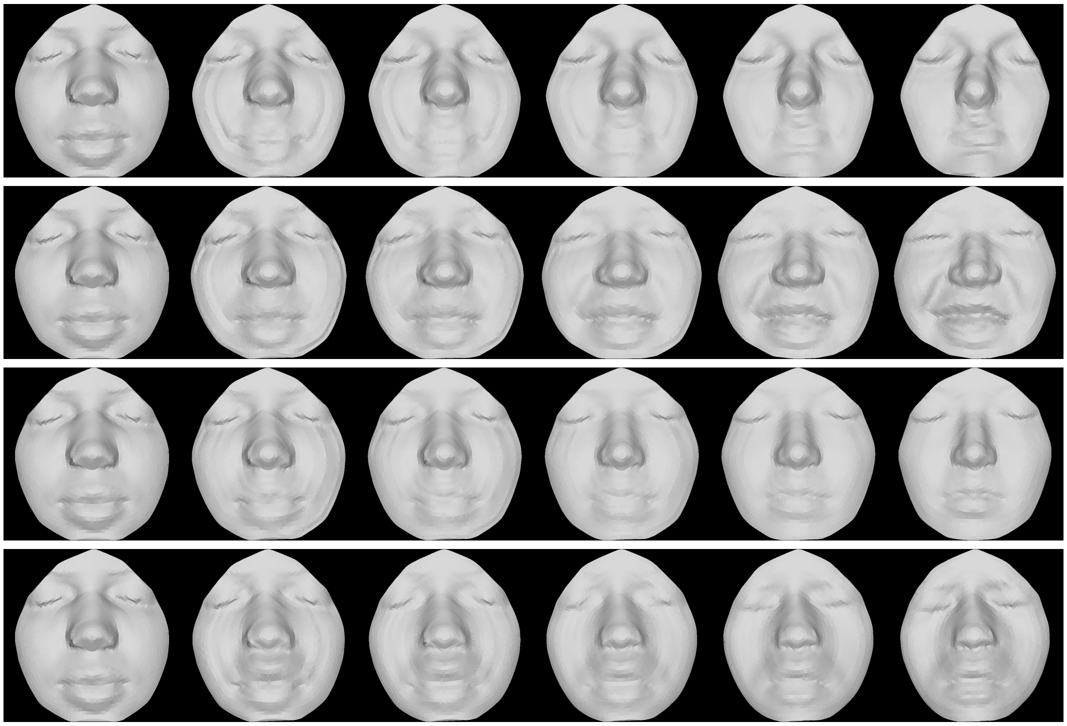


Fig. 11 Geodesic paths between facial surfaces of different people.

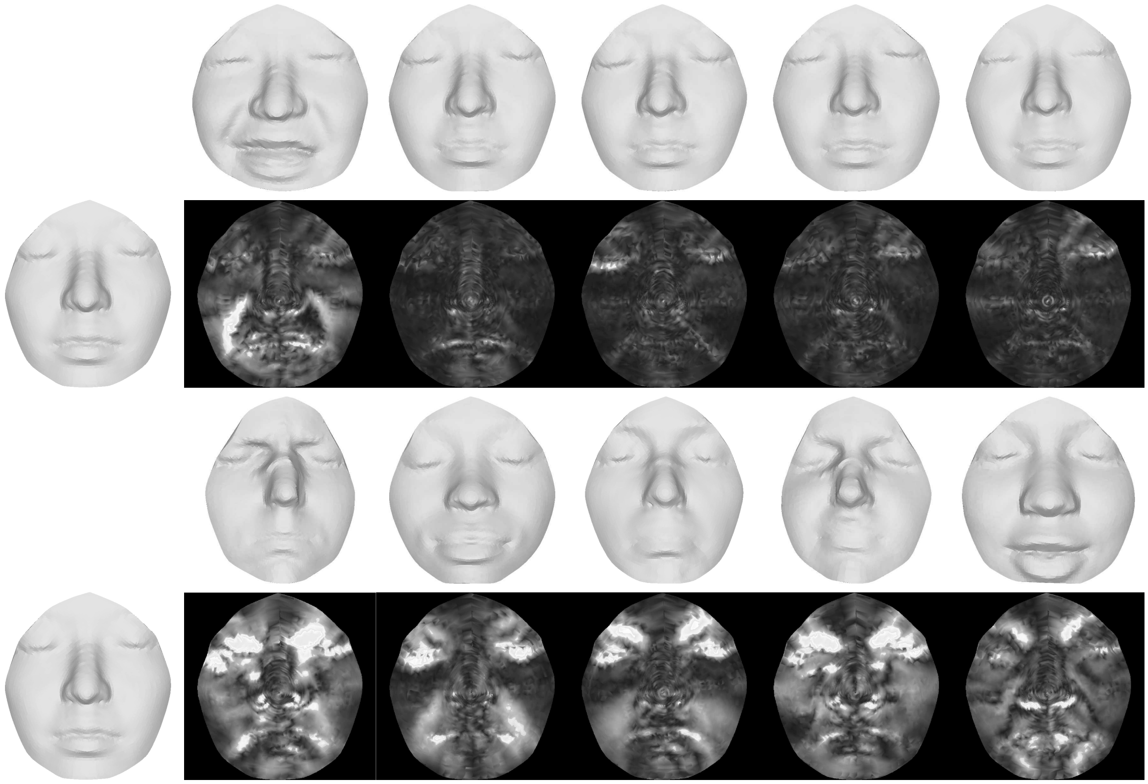


Fig. 12 Magnitude of deformations resulting from geodesic paths shown as scalar fields. The upper row shows the deformations between the same person under different expressions, while the bottom row is for different persons. In each case, we use the same source face, but the target faces are different.

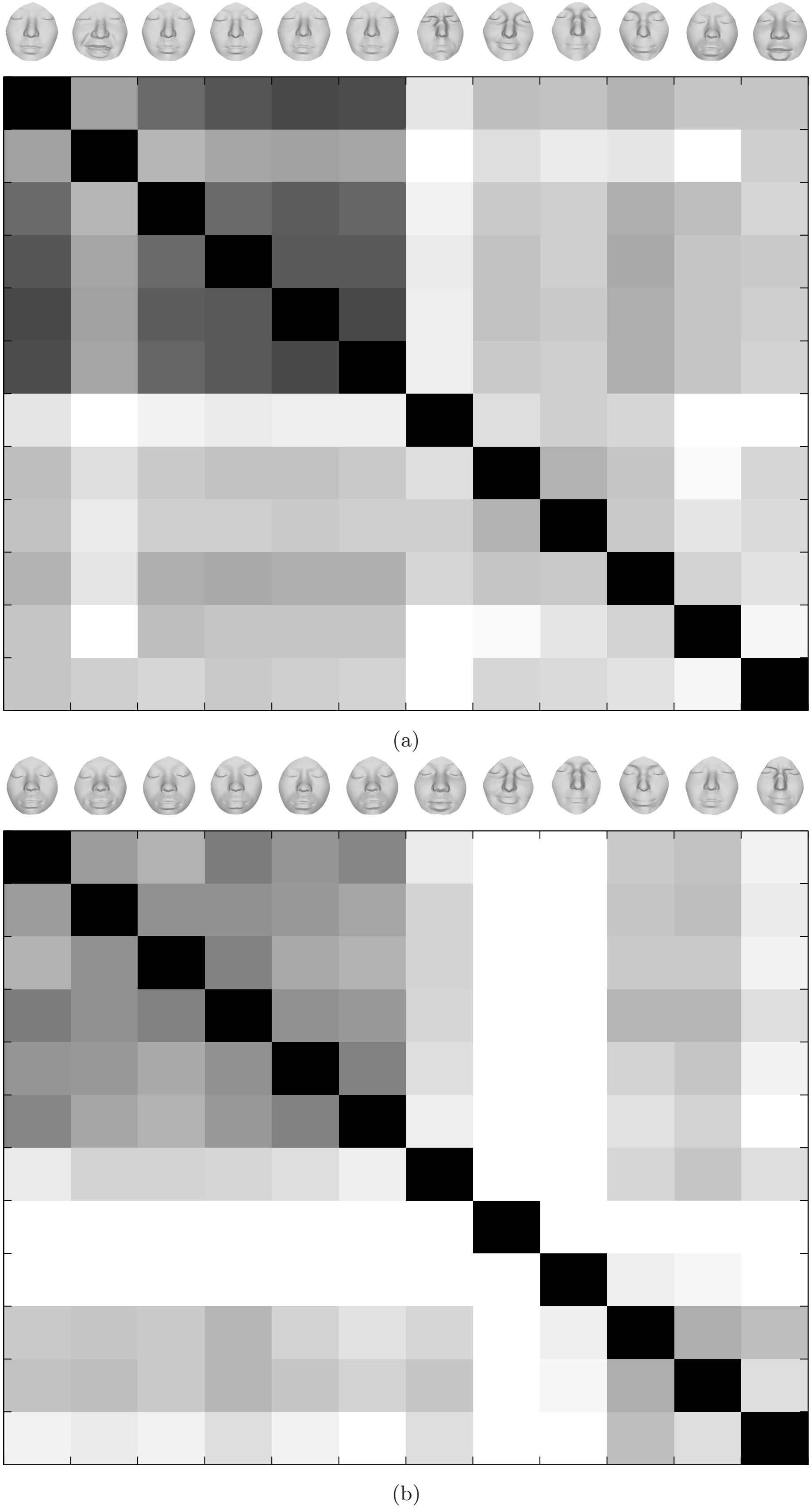
Algorithm 3 : Gradient search

```
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 $i = 1, ..., n$ choose the tangent vector $f_i \in T_{\mu_k}(\mathcal{H})$ which is tangent to the geodesic from $\mu_k$ to $S_i$, and whose norm is equal to the length of this shortest geodesic. The vector $g = \sum_{i=1}^{i=n} f_i$ is proportional to the gradient at $\mu_k$ of the function $\mathcal{V}$.
  2- Flow for time $\epsilon$ along the geodesic which starts at $\mu_k$ and has velocity vector $g$. Call the point where you end up $\mu_{k+1}$, i.e. $\mu_{k+1} = \Psi(\mu_k; \epsilon; g)$.
  3- Set $k = k + 1$ and go to step 1.
```

is far, in terms of the geodesic distance, to each of the faces, and appears to be a blurred version of the individual faces.

6 Extensions to More General Surfaces

The framework proposed in this paper can be applied to more general surfaces if there is a natural way of representing them as indexed collections of closed curves. There is an emerging literature of studying shapes of surfaces by defining a certain Morse function on them and to generate Reeb graphs using the critical points of that function [30]. Our approach would be to use the level curves of those Morse functions to develop natural representations of



(a)

(b)

Fig. 13 Two examples of matrices of pairwise distances between 12 facial surfaces in $\mathcal{H}$. In each case, the first six belong to the same person while the next six belong to six different persons.

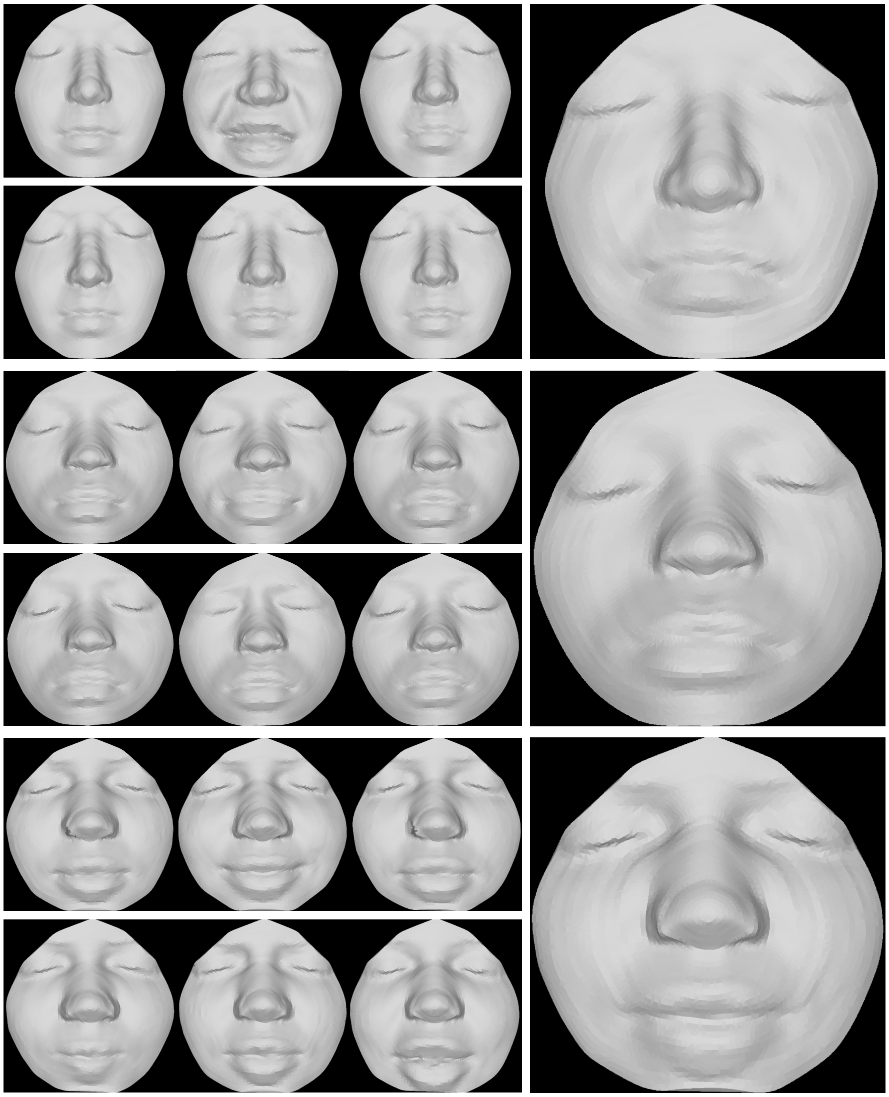


Fig. 14 The same person, at each row, under six facial expressions in the left panel, and the corresponding Karcher mean in the right panel.

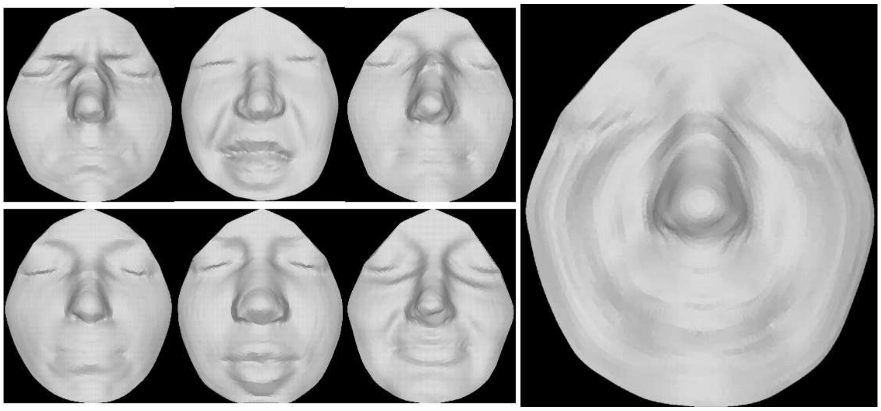


Fig. 15 Karcher mean (right) for the six different faces in the left.

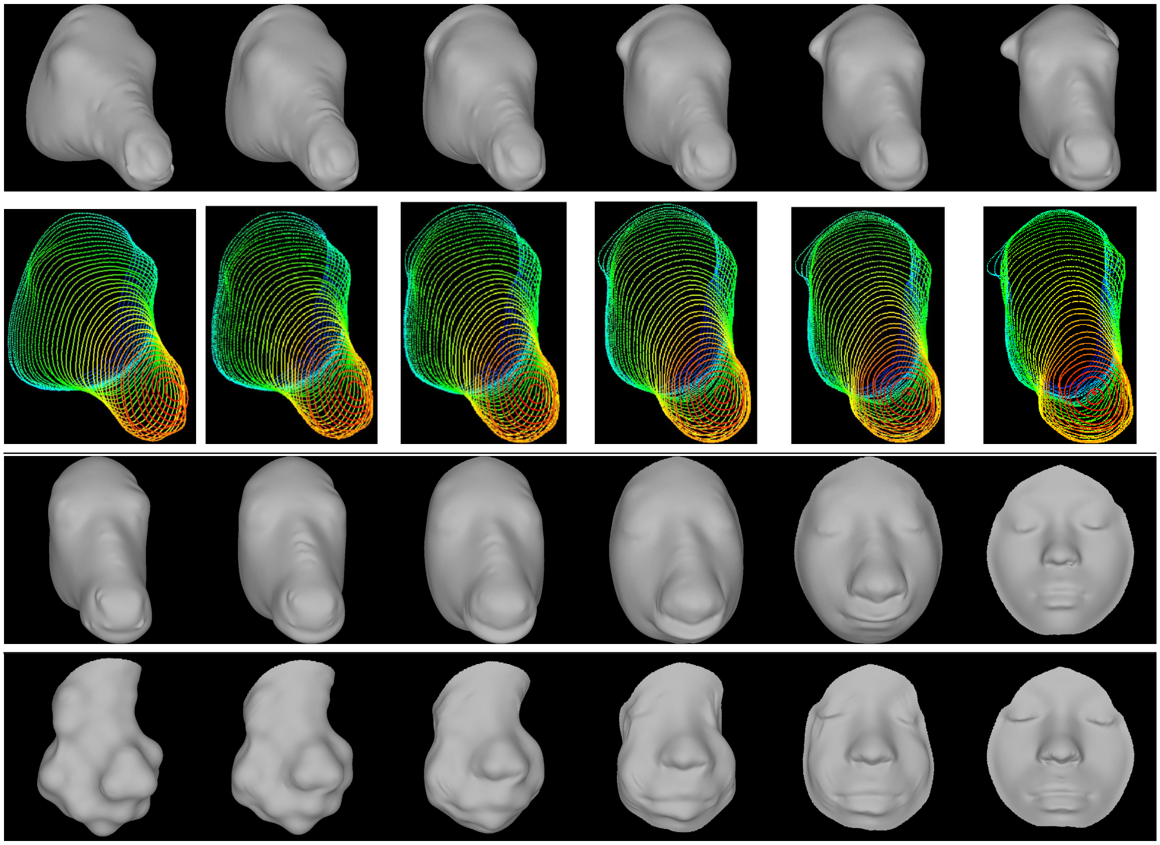


Fig. 16 Examples of geodesic-based deformations between more general surfaces. In the first case, we show both the surfaces and the level curves. while for the next two cases we show only the surfaces.

surfaces. If one wants to restrict to the surface distance function, as in our approach for faces, it becomes important to define a reference point with respect to which *dist* will be defined. If this step is feasible for a surface, the rest of our approach applies directly. Shown in Figure 16 are some illustrative results on computing geodesics between more general kinds of surfaces. The top row shows a geodesic path between the heads of a cow and a horse, and the second row shows the same path but each surface is now displayed as a collection of curves (in $\mathcal{H}$). The third row is a geodesic between the head of a cow and a human face, and the bottom row is a geodesic between a generic blob and a face. In these examples, the reference points for the non-facial surfaces were picked manually but the remainder of the procedure is completely automatic.

7 Summary

In this paper we have introduced a new computational framework for analyzing shapes of facial surfaces. The basic idea is to choose a specific representation of facial surfaces that is composed of level curves of the surface distance function (measured from the tip of nose). Each surface is represented as a path on the space of closed curves in $\mathbb{R}^3$. Focusing on the set of such paths, we impose a Riemannian structure on it that measures the amount of work done in bending one surface into another. The main computational tool presented in this paper is the construction of geodesic paths between arbitrary two facial surfaces in the aforementioned set. There are multiple applications 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 the 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. An extension of this framework to more general surfaces is also presented. This extension assumes that for these surfaces it is possible to define reference points and distance functions so that the surfaces can be represented by level curves.

The geodesics between facial surfaces, or more explicitly facial curves, are computed using the idea of path-straightening [13] under a bending-only metric. 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 [17]. In a more recent paper we have proposed to apply the framework of elastic curves to analyze facial curves [27].

Acknowledgments

The research presented here was supported by in part the grants ARO W911NF-04-01-0268, AFOSR FA9550-06-1-0324, and Northrop-Grumman Innovation Alliance grant to Anuj Srivastava.

Appendices

Note: To apply this appendix to the proof of Proposition 1, let $M = \mathbb{R}^3 \times \mathcal{C} \times \mathbb{R}_+$ and let $\mathcal{F} = \mathcal{H}$.

Let M be a Riemannian manifold and denote by I the unit interval $[0, 1]$. Also, denote by $\mathcal{F}$ the space of measurable functions $[0, L] \to M$. $\mathcal{F}$ is a manifold; its tangent space is given as follows. If $\alpha \in \mathcal{F}$, then

$$T_\alpha(\mathcal{F}) = \left\{ w : [0, L] \to TM : \forall \lambda \in [0, L], w(\lambda) \in T_{\alpha(\lambda)}(M) \text{ and } \int_0^L \langle w(\lambda), w(\lambda) \rangle \, d\lambda < \infty \right\}$$

In other words, 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

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

where the inner product inside the integral uses the Riemannian metric on M.

Theorem 1 *Suppose we are given a path in $\mathcal{F}$ represented as $\phi : [0, L] \times I \to \mathcal{F}$. For each $\lambda \in [0, L]$, define $\phi_\lambda : I \to M$ 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 M.*

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

$$\mathcal{B} = \{\gamma : I \to M \mid \gamma \text{ is smooth, } \gamma(0) = v_0 \textit{ and } \gamma(1) = v_1\}.$$

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

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

Then, γ is a geodesic in M, 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 [26].

Now suppose that $\tilde{\phi} : [0, L] \times I \times (-\epsilon, \epsilon) \to M$ 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 $\tilde{\phi}(., ., h)$ in $\mathcal{F}$ as follows:

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

Differentiating with respect to h at h = 0 gives:

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

If we assume that ϕ_λ is a geodesic in M 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. F. Cazals, J-C Faugère, M. Pouget, and F. Rouillier. The implicit structure of ridges of a smooth parametric surface. *Comput. Aided Geom. Des.*, 23(7):582–598, 2006.
3. K. I. Chang, K. W. Bowyer, and P. J. Flynn. An evaluation of multimodal 2D+3D face biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):619–624, 2005.
4. K. I. Chang, K. W. Bowyer, and P. J. Flynn. Multiple nose region matching for 3D face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695– 1700, 2006.
5. E. W. Dijkstra. *A note on two problems in connection with graphs.* Numerische Mathematik, 1959.
6. H. Edelsbrunner. *Geometry and Topology for Mesh Generation.* Cambridge University Press, England, 2001.
7. J. Glaunes, A. Trouvé, and L. Younes. Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching. In *IEEE Conference on Computer Vision and Pattern Recognition (2)*, pages 712–718, 2004.
8. P.W. Hallinan, G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford. *Two- and Three-Dimensional Patterns of Face.* A. K. Peters, 1999.
9. M. Hilaga, S. T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM SIGGRAPH*, page 203212, 2001.

10. Varun Jain, Hao Zhang, and Oliver van Kaick. Non-rigid spectral correspondence of triangle meshes. *International Journal on Shape Modeling*, 13(1):101–124, 2007.
11. David G. Kendall. Shape manifolds, procrustean metrics and complex projective spaces. *Bulletin of London Mathematical Society*, 16:81–121, 1984.
12. 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.
13. E. Klassen and A. Srivastava. Geodesic between 3D closed curves using path straightening . In A. Leonardis, H. Bischof, and A. Pinz, editors, *European Conference on Computer Vision*, pages 95–106, 2006.
14. E. Klassen, A. Srivastava, W. Mio, and S. Joshi. Analysis of planar shapes using geodesic paths on shape spaces. *IEEE Pattern Analysis and Machine Intelligence*, 26(3):372–383, March, 2004.
15. F. Memoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. Comput. Math.*, 5(3):313–347, 2005.
16. P. W. Michor and D. Mumford. Riemannian geometries on spaces of plane curves. *Journal of the European Mathematical Society*, 8:1–48, 2006.
17. W. Mio, A. Srivastava, and S. Joshi. On shape of plane elastic curves. *International Journal of Computer Vision*, 73(3):307 – 324, July 2007.
18. R. Osada, T. Funkhouser, B. Chazells, and D. Dobkin. Matching 3D models with shape distributions. In *IEEE Shape Modeling International*, May 2001.
19. X. Pennec. Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127–154, 2006.
20. R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233, 2007.
21. C. Samir, A. Srivastava, and M. Daoudi. Three-dimensional face recognition using shapes of facial curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1847–1857, 2006.
22. C. Samir, A. Srivastava, and M. Daoudi. Human face recognition using 2D facial curves. In *International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Toulouse, France*, volume 5, May, 2006.
23. 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.
24. 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.
25. Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien. Surface coding based on morse theory. *IEEE Computer Graphics and Applications*, 11:67–78, 1991.
26. Michael Spivak. *A Comprehensive Introduction to Differential Geometry, Vol I & II.* Publish or Perish, Inc., Berkeley, 1979.
27. A. Srivastava, C. Samir, S. H. Joshi, and M. Daoudi. Elastic shape models for facial shape analysis using curvilinear coordinates. *Journal of Mathematical Imaging and Vision*, accepted for publication, 2008.
28. 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.
29. J.-P. Thirion and A. Gourdon. The 3D marching linesalgorithm and its application to crest lines extraction. INRIA Technical Report 1672, INRIA, Rocquencourt, 1992.
30. J. Tierny, J.-P. Vandeborre, and M. Daoudi. Enhancing 3D mesh topological skeletons with discrete contour constrictions. *The Visual Computer*, to appear, 2007.
31. M. Vaillant and J. Glaunes. Surface matching via currents. In *Proceedings of IPMI*, pages 381–392, 2005.
32. S. Wang, Y. Wang, M. Jin, Xianfeng Gu, and Dimitris Samaras. 3D surface matching and recognition using conformal geometry. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2453–2460, 2006.
33. 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.
34. L. Younes. Computable elastic distance between shapes. *SIAM Journal of Applied Mathematics*, 58:565–586, 1998.