Coordinate descent optimization for one-to-one correspondence and supervised classification of 3D shapes

Chafik Samir, Wen Huang

Version of Record: https://www.sciencedirect.com/science/article/pii/S0096300320304951
Manuscript_b088f2f5237f60adb80f8f6a4bc10536

Coordinate Descent Optimization for One-to-One Correspondence and Supervised Classification of 3D Shapes

Chafik Samir[a,*], Wen Huang[b,*]

[a]*CNRS-UCA, Institut de Mathématiques de Toulouse, France*
[b]*School of Mathematical Sciences, Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian, P.R.China, 361005*

Abstract

Recent developments in shape analysis and retrieval play an important role in a wide variety of applications that potentially require matching of objects with different geometries. In shape classification, there is no natural way to represent an object but the similarity measure, distance between representations or descriptors, depends heavily on the strategy of computing optimal correspondences. In this paper we introduce a new numerical method for registering surfaces. Thus, finding an optimal one-to-one correspondence between their shapes. Unfortunately, solving this type of optimization problem is generally hard because the solutions space is nonlinear with no natural manifold structure on it. To overcome such limitations we make use of recent methods to represent objects then we find optimal correspondences using a discretized approximation of the search space. The proposed method has the advantage of extending the Riemannian analysis of 3D curves in a natural way for surfaces. We demonstrate the proposed algorithms using different public datasets for 2D and 3D objects matching and classification.

Keywords: `Optimization on manifolds ; Coordinate descent ; Parameterized surfaces registration ; Shape classification ; Manifold Learning`

1. Introduction

Due to the importance of finding optimal correspondences among 2D and 3D shapes, there is a great deal of research, analysis and methods that have been introduced for the same purpose [1, 2, 3]. Despite all significant advances that have been achieved (particularly for 2D shapes), the problem of a complete matching between 3D objects is far from being solved. Basically, the main challenges are related to data and the application at hand. At the same time, planar closed curves or boundary surfaces are increasingly being used to represent the outline or shape of objects, enabling them to emerge as a natural choice in shape analysis and their applications. The need for shape analysis arises in many scientific fields, including computer vision, medical imaging, computer graphics, among others [4, 5, 6]. Analyzing shapes and their differences relies on the notion of distance between shapes which is the key element for registration, deformation, and comparison of shapes. For example, in medical imaging, surface registration is always needed for constructing generative models, shape classification and prediction, and supervised segmentation, morphometry, fusion, denoising observed data [7, 8, 9]. In many cases, a surface must match another, while comparing them. Finding an optimal registration that best matches the required constraints is difficult in practice, especially on surfaces undergoing nonlinear deformations [10]. It is therefore necessary to explore efficient numerical methods

*Corresponding authors
Email addresses: `chafik.samir@uca.fr` (Chafik Samir), `wen.huang@xmu.edu.cn` (Wen Huang)

to compute optimal re-parametrization of parametrized surfaces. There are several successful approaches for aligning two point clouds, matching non-rigid shapes [11], registering curves to surfaces as a partial matching [12], etc. In this paper, we consider a different problem: Find an optimal one-to-one correspondence between parameterized tubular surfaces as a Riemannian optimization on the space of all re-parametrizations.

Solving this type of optimization problem is generally hard, since the space of all re-parametrizations is a complicated infinite dimensional and nonlinear functional space [13]. Considering the actions of the group of diffeomorphisms on the simplest closed manifold, namely the unit circle, of the apparent simplicity, it has been shown that it is highly nontrivial and very extensive [14, Chapter 3]. Thus, developing an efficient optimization method to register two parametrized surfaces modulo all re-parametrizations is a serious challenge, and a loss of quality is often observed during this step. To overcome such limitations, at least numerically, one can: *i)* choose a good representation for surfaces, *ii)* build a metric space of those representations, *iii)* formulate the registration problem as a minimization of a cost function, and finally *iv)* discretize the space of re-parameterization and optimize over a finite dimensional space [15, 16].

Optimization procedures as a part of shape analysis play a crucial role in many applications making it an active research area. That is why continuous and discrete optimization procedure for shape analysis has been extensively studied during last decades [17, 18]. Previous methods usually compute the solution by minimizing certain energy functionals (e.g. variational formulations). However, the literature on parameterization of tubular objects is by far not as extensive as for open and spherical surfaces. We briefly describe some related methods, a non-exhaustive list, that have been used for similar contexts. For example paper [19] developed a framework to compute large deformations diffeomorphisms between surfaces but their framework based on manually selected points is not applicable for parametrized surfaces. An extension has been presented in [6]. A more complete analysis of shapes of parametrized surfaces has been presented in [20, 21] and the theoretical solutions are considered over the set of all diffeomorphisms and represented by smooth vector fields. However optimization procedures in general are either restrictive to a parametric family of wrapping functions or slow with unsatisfying minimizers in practice.

In this work we present a new solution for parametrized tubular surfaces and we show later that this framework can be applied on other surfaces. Finding a one-to-one correspondence is a fundamental problem in mathematics and computation [22, 23, 24] with applications in multiple fields, ranging from medical imaging [25] to computer vision [26], and passing by Computer Aided Design (CAD) [27]. This framework is motivated by shape classification as a supersized learning from a training set of data containing 2D curves (or tubular surfaces) whose classes membership is known. In particular and using new algorithms we show that the accuracies of some standard classifiers are better when the one-to-one correspondence is optimal.

A tubular object is defined to be a surface that is homomorphic to a cylinder, or equivalently, is defined to be a differentiable function $S : \Omega \in \mathbb{R}^3 : (\xi_1, \xi_2) \mapsto S(\xi_1, \xi_2)$, where Ω denotes $[0, 1] \times \mathbb{S}^1$ and $\mathbb{S}^1$ denotes the unit circle. The correspondence can be characterized mathematically by a diffeomorphism. The set of diffeomorphisms, or equivalently called the re-parameterizations group, is defined by

$$\Gamma(\mathbb{D}) = \{\gamma : \mathbb{D} \to \mathbb{D} \,|\gamma \text{ is an orientation-preserving} \\ \text{diffeomorphism on the domain } \mathbb{D}.\}$$

In particular, for the tubular object, the domain $\mathbb{D}$ is Ω and the boundary is preserved with $\gamma(\partial\Omega) = \partial\Omega$. In this setting, looking for a reparameterization leads to an optimization problem:

$$\min_{\gamma \in \Gamma([0,1]\times\mathbb{S}^1)} E(S_1, S_2 \circ \gamma), \tag{1}$$

where E is a real-valued cost function and is usually a distance function in the space of surfaces, and S_1 and S_2 are two tubulars.

Solving the optimization problem (1) efficiently is not easy. For shape of curves, the problem is relatively simple, and Riemannian optimization method [28], dynamic programming [16] and coordinate descent have

been proposed. These methods have the ability to escape from unsatisfied local minimizers, see their papers for details. However, the existing methods for the problem (1) is still limited. To the best of our knowledge, only the steepest descent method with multiple initial iterates have been used. However, gradient-based algorithms are usually slow and unsatisfied since i) the steepest descent is known to be slow with a local behavior in nonlinear optimization problems and ii) the number of initial iterates need to be large due to the large dimension of the domain (infinite in theory and large in practice).

Contributions. We propose a new method to solve this problem for tubular surfaces with an extension to other surfaces. For purposes of simplicity, we often use the term surface. The main contributions of this work are:

1. We use a recent work for elastic shape analysis of surfaces [21] and build a connection with the framework of elastic shape analysis of curves in [16], see Proposition 3.1.
2. We propose a new optimization method (algorithm 2 and algorithm 3) to register surfaces.
3. We use various examples with complicated geometries for illustrations and different benchmarks to show the effectiveness of the new methodology. The 3D shape classification is carried out by combining the proposed method for correspondence and a Gaussian process autoencoder for classification.

This paper is organized as follows. Section 2 reviews a framework of elastic shape analysis of curves and a recent framework of elastic shape analysis of surfaces. Section 3 shows a relationship between the two frameworks. This motivates the algorithm design and yields Section 4, which presents the proposed optimization algorithms. Section 5 shows that the idea of using curves for surfaces can be applied for other surfaces. The numerical illustrations and experiments are given in Section 6 to show the accuracy of the proposed algorithms. Finally, the conclusion is carried out in Section 7.

2. Elastic Shape Analysis of Curves and Surfaces

Intuitively, a shape is invariant to translation, scaling, rotation and re-parameterization. In elastic shape analysis, a mathematical representation of shape is defined such that it is invariant to the four operations above, and multiple frameworks of elastic shape analysis have been proposed, see e.g., [29, 16, 21]. This paper focuses on the removing reparameterization, which is the most difficult one to be removed. We refer to [16] and [21] if readers are interested in removing the other three operations.

2.1. Analyzing Curves

In this paper, the SRVF framework defined in [16] is considered. Specifically, a curve in $\mathbb{R}^n$ is denoted by a parameterized function $\beta : \mathbb{D} \to \mathbb{R}^n$, where $\mathbb{D}$ is $[0, 1]$ for open curves and $\mathbb{D}$ is the unit circle $\mathbb{S}^1$ for closed curves. The square root velocity function of β is

$$q(t) = \begin{cases} \frac{\dot{\beta}(t)}{\sqrt{||\dot{\beta}(t)||_2}}, & \text{if } ||\dot{\beta}(t)||_2 \neq 0; \\ 0, & \text{if } ||\dot{\beta}(t)||_2 = 0. \end{cases} \tag{2}$$

where $\|\cdot\|_2$ denote the 2-norm. The curve β can be recovered by $\beta(t) = \int_0^t q(s)||q(s)||_2 ds + \beta_0$, where $\beta_0 \in \mathbb{R}^n$ is the starting point of β.

The group action on curve $(\beta, \gamma) = \beta \circ \gamma$ is equivalent to the action $(q, \gamma) = \sqrt{\dot{\gamma}} q \circ \gamma$ on q. Let $\bar{\mathcal{N}}$ denote the space of all q functions. For open curves, $\bar{\mathcal{N}}$ is the $\mathbb{L}^2$ space. For closed curves, $\bar{\mathcal{N}} = \{q \in \mathbb{L}^2 \mid \int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0\}$ is a subspace of the $\mathbb{L}^2$ space. Note that the condition $\int_{\mathbb{S}^1} q(t)\|q(t)\|_2 dt = 0$, equivalent to $\int_{\mathbb{S}^1} \dot{\beta}(t) dt = 0$, implies that the starting point and end point of the curve β is the same and therefore is a closed curve. A quotient space of $\bar{\mathcal{N}}$ can be defined as $\mathcal{N} = \{[q] \mid q \in \bar{\mathcal{N}}\}$, where $[q] = \{(q, \gamma) \mid \gamma \in \Gamma(\mathbb{D})\}$ is an orbit. It has been shown in [16] that the group action on q is isometric with respect

to the $\mathbb{L}^2$ metric. Therefore, $\mathcal{N}$ with the $\mathbb{L}^2$ metric is a well defined metric space. The cost function for finding the best reparameterization between two curves therefore is (see [16])

$$\begin{aligned} C : \Gamma(\mathbb{D}) \to \mathbb{R} : \gamma \to & \|q_1 - (q_2, \gamma)\|^2_{\mathbb{L}^2} \\ & = \int_{\mathbb{D}} \|q_1(t) - \sqrt{\dot{\gamma}} q_2 \circ \gamma(t)\|_2^2 dt \end{aligned} \tag{3}$$

2.2. Analyzing Surfaces

We first give a definition for axial and circular curves.

Definition 2.1. *Let S be a surface parametrized with (ξ_1, ξ_2) such that: $S : (\xi_1, \xi_2) \in [0,1] \times \mathbb{S}^1 \mapsto S(\xi_1, \xi_2) \in \mathbb{R}^3$. We adapt the following definitions:*

1. *ξ_1-curves are axial curves on S with ξ_2 constant.*
2. *ξ_2-curves are circular curves on S with ξ_1 constant.*

Suppose $S : [0,1] \times \mathbb{S}^1 \to \mathbb{R}^3$ is a differentiable surface. Let $\frac{\partial S}{\partial \xi_1}$ and $\frac{\partial S}{\partial \xi_2}$ be the two partial derivatives of S along the ξ_1-curves and ξ_2-curves. A normal field on S with respect to a positive orientation is defined as follows:

$$N : \quad \Omega \to \mathbb{R}^3 : (\xi_1, \xi_2) \mapsto N(\xi_1, \xi_2) = \frac{\partial S}{\partial \xi_1} \times \frac{\partial S}{\partial \xi_2},$$

where $\times$ denotes the cross product. The Q-field of S is defined as Square Root Normal Field(SRNF) [21]:

Definition 2.2. *Let S be a surface with a parametrization (ξ_1, ξ_2) we call Q-field of S the vector field on S along ξ_1 and ξ_2 defined by:*

$$\begin{aligned} Q : & \Omega \to \mathbb{R}^3 \\ & (\xi_1, \xi_2) \mapsto Q(\xi_1, \xi_2) = \frac{N(\xi_1, \xi_2)}{\sqrt{\|N(\xi_1, \xi_2)\|_2}} \end{aligned} \tag{4}$$

One can see that the framework of SRNF is similar to the framework of SRVF in the sense that SRNF considers taking the square root of the normal field instead of the velocity. This is a natural generalization from 1D curves to 2D surfaces. It is probably the reason that analyzing a curve segment on a 2D surface by SRNF is essentially using the SRVF, see details in Section 3.

Proposition 2.1 gives the group action on Q-fields and proves the isometry of the group action under $\mathbb{L}^2$ metric. These results have been given for spherical surfaces in [21]. We give them for tubular surfaces and for completeness.

Proposition 2.1. *Let S be a surface and Q its corresponding Q-field, we have the following properties:*

1. *The group action on surface $(S, \gamma) = S \circ \gamma$ is equivalent to $(Q, \gamma) = \sqrt{|J_\gamma|} Q \circ \gamma$, where J_γ is the Jacobi of $\gamma \in \Gamma(\Omega)$ and $|\cdot|$ denotes the determinant.*
2. *The group action of γ on Q-fields is an isometry under the $\mathbb{L}^2$ metric.*

Specifically, we have

$$\begin{aligned}
\|(Q_1,\gamma)-(Q_2,\gamma)\|^2_{\mathbb{L}^2} &= \left\|\sqrt{|J_\gamma|}Q_1\circ\gamma-\sqrt{|J_\gamma|}Q_2\circ\gamma\right\|^2_{\mathbb{L}^2}\\
&= \int_\Omega \left\|\sqrt{|J_\gamma|}Q_1\circ\gamma(\xi_1,\xi_2)-\sqrt{|J_\gamma|}Q_2\circ\gamma(\xi_1,\xi_2)\right\|^2_2 d\xi_1 d\xi_2\\
&= \int_\Omega |J_\gamma|\,\|(Q_1-Q_2)\circ\gamma(\xi_1,\xi_2)\|^2_2\, d\xi_1 d\xi_2\\
&= \int_\Omega |J_\gamma||J_{\gamma^{-1}}|\,\|(Q_1-Q_2)(\xi_1,\xi_2)\|^2_2\, d\xi_1 d\xi_2\\
&= \|Q_1-Q_2\|^2_{\mathbb{L}^2}
\end{aligned}$$

The space of Q-fields, denoted by $\bar{\mathcal{M}}$, is

$$\left\{Q\in\mathbb{L}^2 \mid \int_{\mathbb{S}^1} Q(\xi_1,t)\|Q(\xi_1,t)\|_2 dt = 0 \text{ for all } \xi_1\in[0,1]\right\}, \tag{5}$$

and $\mathcal{M}$ denote the quotient space $\{[Q] \mid Q\in\bar{M}\}$, where $[Q]$ denotes the orbit of Q under the group action $\Gamma(\Omega)$, i.e., $[Q]=\{(Q,\gamma)\mid\gamma\in\Gamma(\Omega)\}$. Since the group action is isometric under the $\mathbb{L}^2$ metric, $\mathcal{M}$ with $\mathbb{L}^2$ is a well-defined metric space.

Given two surfaces S_1 and S_2, let Q_1 and Q_2 denote their Q-fields, respectively. We define the distance between Q_1 and Q_2 to be $\|Q_1-Q_2\|_{\mathbb{L}^2}$, since the space of Q is a subspace of $\mathbb{L}^2$. The best reparameterization $\hat{\gamma}$ is therefore the minimizer of the functional:

$$\begin{aligned}
E:&\Gamma(\Omega)\to\mathbb{R}:\gamma\mapsto\|Q_1-(Q_2,\gamma)\|^2_{\mathbb{L}^2}\\
&= \int_\Omega \left\|Q_1(\xi_1,\xi_2)-(Q_2,\gamma)(\xi_1,\xi_2)\right\|^2_2 d\xi_1 d\xi_2,
\end{aligned} \tag{6}$$

where $(Q_2,\gamma)(\xi_1,\xi_2)=\sqrt{|J_\gamma|}Q_2\circ\gamma(\xi_1,\xi_2)$.

3. Optimal Parametrization Between Surfaces

Optimizing over a 2-dimensional domain is usually difficult and expensive due to: *i*) it is relatively simple to parametrize a curve by arc-length using numerical integration, whereas there is no natural way (or directions) to parametrize a surface, and *ii*) the space all re-parametrizations is infinite dimensional function space (or a group) with no natural manifold structure. In this section, we exploit the fact that a surface can be parametrized by a collection of axial and circular curves to show that the cost function (6) can be reduced to a set of the cost function of curves using the SRVF representation.

Proposition 3.1 gives an important fact that motivates Algorithm 2 for finding optimal correspondences between two surfaces.

Proposition 3.1. *Let $\gamma \in \Gamma(\Omega)$, denoted by $\gamma : [0,1] \times \mathbb{S}^1 \to [0,1] \times \mathbb{S}^1 : (\xi_1, \xi_2) \mapsto (\gamma_1(\xi_1, \xi_2), \gamma_2(\xi_1, \xi_2))$, where $\gamma_1 : [0,1] \times \mathbb{S}^1 \to [0,1]$ and $\gamma_2 : [0,1] \times \mathbb{S}^1 \to \mathbb{S}^1$. Given two surfaces S_1 and S_2 and their corresponding Q-fields Q_1 and Q_2:*

- *Let $\alpha_\lambda(\xi_2)$ denote $\gamma_2(\lambda, \xi_2)$ and therefore $\alpha_\lambda : \mathbb{S}^1 \to \mathbb{S}^1$. Suppose $\gamma_1(\xi_1, \xi_2) = \xi_1$ for all $\xi_2 \in \mathbb{S}^1$. Then for any given $\lambda \in [0,1]$, the function*

$$h_\lambda : \Gamma(\mathbb{S}^1) \to \mathbb{R} : \alpha_\lambda \mapsto h_\lambda(\alpha_\lambda) = \int_{\mathbb{S}^1} \left\| Q_1(\lambda, \xi_2) - (Q_2, \gamma)(\lambda, \xi_2) \right\|_2^2 d\xi_2$$

  can be reformulated into the form of (3)*.*

- *Likewise, let $\beta_\lambda(\xi_1)$ denote $\gamma_1(\xi_1, \lambda)$, and therefore $\beta_\lambda : [0,1] \to [0,1]$. If $\gamma_2(\xi_1, \xi_2) = \xi_2$ holds for all $\xi_1 \in [0,1]$, the function*

$$k_\lambda : \Gamma([0,1]) \to \mathbb{R} : \beta_\lambda \mapsto k_\lambda(\beta_\lambda) = \int_0^1 \left\| Q_1(\xi_1, \lambda) - (Q_2, \gamma)(\xi_1, \lambda) \right\|_2^2 d\xi_1$$

  also can be reformulated into the form of (3)*.*

Proof. First of all, using the assumption of γ, the determinant of the Jacobi of γ is

$$\begin{aligned} \det\left(\begin{bmatrix} \frac{d}{d\xi_1}\gamma_1 & \frac{d}{d\xi_2}\gamma_1 \\ \frac{d}{d\xi_1}\gamma_2 & \frac{d}{d\xi_2}\gamma_2 \end{bmatrix} \right) &= \det\left(\begin{bmatrix} 1 & 0 \\ \frac{d}{d\xi_1}\gamma_2 & \frac{d}{d\xi_2}\gamma_2 \end{bmatrix} \right) \\ &= \frac{d}{d\xi_2}\gamma_2. \end{aligned}$$

Let $q_\lambda(\xi_2)$ and $p_\lambda(\xi_2)$ denote $Q_1(\lambda, \xi_2)$ and $Q_2(\lambda, \xi_2)$, respectively. We have

$$\begin{aligned} h_\lambda(\alpha_\lambda) &= \int_{\mathbb{S}^1} \left\| Q_1(\lambda, \xi_2) - (Q_2, \gamma)(\lambda, \xi_2) \right\|_2^2 d\xi_2 \\ &= \int_{\mathbb{S}^1} \left\| Q_1(\lambda, \xi_2) - \sqrt{|J_\gamma|}(Q_2 \circ \gamma)(\lambda, \xi_2) \right\|_2^2 d\xi_2 \\ &= \int_{\mathbb{S}^1} \left\| q_\lambda(\xi_2) - \sqrt{\dot{\alpha_\lambda}} p_\lambda \circ \alpha_\lambda(\xi_2) \right\|_2^2 d\xi_2, \end{aligned}$$

which completes a proof for h. Likewise we obtain a proof for k. □

4. Optimization Algorithms

In Section 4.1, we present an optimization algorithm for a general reparameterization problem on $\Gamma(\Omega)$. An implementation by discretization is described in Section 4.2 and a technique for handling large deformations is discussed in Section 4.3.

4.1. Optimization over $\Gamma(\Omega)$

The optimization problem (6) can be written as

$$\min_{\gamma\in\Gamma(\Omega)} F(\gamma)$$

Define $\mathcal{N}_1 = \{\gamma \in \Gamma(\Omega) \mid \gamma_1(\xi_1, \xi_2) = \xi_1 \text{ for all } \xi_2 \in \mathbb{S}^1\}$ and $\mathcal{N}_2 = \{\gamma \in \Gamma(\Omega) \mid \gamma_2(\xi_1, \xi_2) = \xi_2 \text{ for all } \xi_1 \in [0, 1]\}$. We use an optimization algorithm that minimizes F alternatively over $\mathcal{N}_1$ and $\mathcal{N}_2$. The algorithm is stated in Algorithm 1.

Algorithm 1 Alternative Optimization

1: Let $F^0 \leftarrow F$, $\gamma^0 = \mathrm{id}$, and $k \leftarrow 0$;
2: **for** $k = 0, 1, 2, \ldots$ **do**
3: $\quad\hat{\gamma}^k = \arg\min_{\gamma\in\mathcal{N}_1} F^k(\gamma)$;
4: $\quad\tilde{\gamma}^k = \arg\min_{\gamma\in\mathcal{N}_2} F^k \circ \hat{\gamma}^k(\gamma)$;
5: $\quad F^{k+1} = F^k \circ \hat{\gamma}^k \circ \tilde{\gamma}^k$;
6: $\quad\gamma^{k+1} = \gamma^k \circ \hat{\gamma}^k \circ \tilde{\gamma}^k$
7: **end for**

Theorem 4.1 shows that either Algorithm 1 makes progress in the sense of reducing function value or the partial derivative of F^k is 0.

Theorem 4.1. *Suppose the solutions $\hat{\gamma}_k$ and $\tilde{\gamma}_k$ in Algorithm 1 exist. Then one of the following statements hold:*

1. *the function value strictly decreases, i.e., $F(\gamma_{k+1}) < F(\gamma_k)$;*
2. *$\hat{\gamma}^k = \mathrm{id}$, $\tilde{\gamma}^k = \mathrm{id}$, i.e., $\gamma^{k+1} = \gamma^k$, $\frac{\partial}{\partial\gamma_1} F^k|_{\gamma=\mathrm{id}} = 0$, and $\frac{\partial}{\partial\gamma_2} F^k|_{\gamma=\mathrm{id}} = 0$.*

Proof. It can be seen that

$$F(\gamma^k) = F^k(\mathrm{id}) \geq F^k(\hat{\gamma}^k) \geq F^k \circ \hat{\gamma}^k(\tilde{\gamma}^k) = F^{k+1}(\mathrm{id}) = F(\gamma^{k+1}),$$

where $F(\gamma^k) = F(\gamma^{k+1})$ if and only if $\hat{\gamma}^k = \mathrm{id}$ and $\tilde{\gamma}^k = \mathrm{id}$.

Suppose $\hat{\gamma}^k = \mathrm{id}$ and $\tilde{\gamma}^k = \mathrm{id}$. It follows that $\gamma^{k+1} = \gamma^k \circ \hat{\gamma}^k \circ \tilde{\gamma}^k = \gamma^k$. Furthermore, note that $\mathrm{id} = \arg\min_{\gamma\in\mathcal{N}_1} F^k(\gamma)$ is equivalent to $\phi = \arg\min_{\gamma_2:\Omega\to\mathbb{R}} F^k(\varphi, \gamma_2)$, where $\varphi(\xi_1, \xi_2) = \xi_1$ for all $\xi_2 \in \mathbb{S}^1$, and $\phi(\xi_1, \xi_2) = \xi_2$ for all $\xi_1 \in [0, 1]$. It follows that $\frac{\partial F^k}{\partial\gamma_2}(\varphi, \phi) = 0$ which is equivalent to $\frac{\partial}{\partial\gamma_2} F^k|_{\gamma=\mathrm{id}} = 0$. Likewise, it holds that $\frac{\partial}{\partial\gamma_1} F^k|_{\gamma=\mathrm{id}} = 0$. □

4.2. Discretization

A discretization version of Algorithm 1 is stated in Algorithm 2 for minimizing (6). Steps 8 and 14, corresponding to the computations of $F^k \circ \hat{\gamma}^k$ and $F^k \circ \hat{\gamma}^k \circ \tilde{\gamma}^k$, are crucial since using them makes the assumptions of γ in Proposition 3.1 hold. Steps 5 and 11 in Algorithm 2, the minimization steps, correspond to Steps 3 and 4 of Algorithm 1, respectively. Multiple methods have been proposed to solve the inner problems in Steps 5 and 11, such as dynamic programming [16] and Riemannian methods [28]. The steepest descent method with multiple initial iterates has been proposed to optimize the cost function E. However, this method usually does not give a satisfactory result due to the difficulties given in the introduction. Existing methods of curves works on a grid in $[0, 1]^2$ rather than the continuous space $\Gamma([0, 1])$. By exhaustively searching over

Algorithm 2 Alternative Optimization for E

Input: Two Q-functions Q_1 and Q_2.

Output: The minimizer $\gamma^* = \begin{bmatrix} \gamma_1^* \\ \gamma_2^* \end{bmatrix}$.

1: Discretize $[0,1]$ by n points $\lambda \in \{0, \ldots, 1\}$, note h_λ the circular curve at level λ and α_λ its parametrization. Then, each surface has n circular curves.
2: Discretize $\mathbb{S}^1$ by m points $\lambda \in \{0, \ldots, 2\pi\}$, note k_λ the axial curve and at level λ, and β_λ its parametrization. This leads to a discretization of Q_1 and Q_2 by $n \times m$ points in R^3
3: Set $k = 0$, $\gamma_1^*(\xi_1, \lambda) = \xi_1$ and $\gamma_2^*(\lambda, \xi_2) = \xi_2$ for all $\lambda \in [0,1]$;
4: **for** All $\lambda \in \{0, \ldots, 1\}$ **do**
5: Optimize h_λ for Q_1 and Q_2 to find the minimizer $\alpha_\lambda^* \in \Gamma(\mathbb{S}^1)$;
6: **end for**
7: Define $\gamma_2(\xi_1, \xi_2) = \alpha_{\xi_1}^*(\xi_2)$;
8: Set $Q_2(\xi_1, \xi_2)$ to be $\sqrt{\frac{d\gamma_2}{d\xi_2}(\xi_1, \xi_2)} Q_2(\xi_1, \gamma_2(\xi_1, \xi_2))$;
9: Set $\gamma_1^*(\xi_1, \xi_2)$ to be $\gamma_1^*(\xi_1, \gamma_2(\xi_1, \xi_2))$ and $\gamma_2^*(\xi_1, \xi_2)$ to be $\gamma_2^*(\xi_1, \gamma_2(\xi_1, \xi_2))$
10: **for** All $\lambda \in \{0, \ldots, 2\pi\}$ **do**
11: Optimize k_λ for Q_1 and Q_2 to find the minimizer $\beta_\lambda^* \in \Gamma([0,1])$;
12: **end for**
13: Define $\gamma_1(\xi_1, \xi_2) = \beta_{\xi_2}^*(\xi_1)$;
14: Set $Q_2(\xi_1, \xi_2) \leftarrow \sqrt{\frac{d\gamma_1}{d\xi_1}(\xi_1, \xi_2)} Q_2(\gamma_1(\xi_1, \xi_2), \xi_2)$;
15: Set $\gamma_1^{*,k}(\xi_1, \xi_2)$ to be $\gamma_1^*(\gamma_1(\xi_1, \xi_2), \xi_2)$ and $\gamma_2^{*,k}(\xi_1, \xi_2)$ to be $\gamma_2^*(\gamma_1(\xi_1, \xi_2), \xi_2)$;
16: If the algorithm converges, then return $\begin{bmatrix} \gamma_1^{*,k} \\ \gamma_2^{*,k} \end{bmatrix}$ and stop. Otherwise, $k = k + 1$ and go to Step 4.

the piece-wise linear functions on the grid with slope constraints, one can find an approximation of a solution. If the grid is fine, then the approximation can be used as a solution, such as dynamic programming [16]. Otherwise, the approximation can be used as an initial iterate for a gradient method, such as Riemannian methods [28].

In practice, every surface is represented by n points along each circular curve (partition of $[0, 2\pi]$) and m points along an axial curve (partition of $[0, 1]$). Moreover, every surface is a grid of $n \times m$ points. Each point represents the intersection between a circular and an axial curve. Because of the discretization, all gridpoints are disjoints and should not cross each other during the update. Note that when using a dynamic programming approach to register curves, the complexity is quadratic to the number of points on a curve. Therefore, we have a time complexity $\mathcal{O}(m \times n^2)$ for reparametrizing circular curves and $\mathcal{O}(n \times m^2)$ for axial curves. For the Riemannian optimization method, it is shown empirically in [28, Section 6.6] that the complexity is approximately linear to the number of points on curves. Therefore, the complexity in Steps 5 and 11 is approximately $O(mn)$. The complexity in interpolations in Steps 8, 9, 14 and 15 is also $O(mn)$. Therefore, the total complexity of Algorithm 2 with Riemannian optimization method is approximately $O(kmn)$, where k is the number of iterations. In our experiments, the maximum number of iterations is fixed to be 100.

4.3. Successive Infinitesimal Variations

The continuous description of Ω involves a space Γ of infinite dimension. Many algorithms based on discretization have difficulties if the deformations between two shapes are large, and therefore, in practice, it is common to model a time-sequence of smaller, more elementary, deformations, rather than a large deformation, see, e.g., [30, 31]. In this paper, we follow the same idea and propose Algorithm 3 based on Algorithm 2. Concretely, the new methodology can be seen as a spatio-temporal discretization scheme when $(t, \gamma^t) \in [0, 1] \times \Gamma$ leads to a space discretization.

In the case of a uniform parametrization (ξ_1, ξ_2) as an equidistant mesh of squares or rectangles, one can compute the initial re-parametrization by solving a one-dimensional problem in ξ_1 direction and subsequently a family of problems in the ξ_2 direction [32]. Consequently, a better strategy would be to re-parametrize the two Q-fields uniformly (by arc-length in each component), fix the parametrization of Q_1 and then search for reparameterization $\hat{\gamma}$ on Q_2 iteratively. Indeed, at each iteration k we consider a small stepsize $\rho \in [0, 1]$, compute Q_ρ in the direction of Q_1, find γ^k between Q_2 and Q_ρ using Algorithm 2, and then update Q_2 with (Q_2, γ^k). Therefore, the final solution is given by $\hat{\gamma} = \gamma^0 \circ \gamma^1 \circ \dots \gamma^N$.

Note that the boundary conditions is checked using a projection P (see Line 6 in Algorithm 3) from $\mathbb{L}^2$ to $\bar{\mathcal{M}}$ to make sure that circular curves are closed and that the optimal infinitesimal re-parametrization is well defined for every iteration. Consequently, using infinitesimal successive variations guarantees the convergence to a diffeomorphic solution [30, Chapter 11]. According to this iterative algorithm, convergence is achieved when no update of $\hat{\gamma}$ is possible which corresponds to the case where the line between Q_1 and $(Q, \hat{\gamma})$ is orthogonal to $[Q_2]$. We illustrate the idea of successive iterations in Figure 1(g): Original Q_1 and Q_2 define two equivalent classes $[Q_1]$ and $[Q_2]$. We fix the parametrization of Q_1 to be the identity and we search for the optimal re-parametrization of Q_2. The algorithm is initialized with (Q_2, Id_Γ), iterates over a set of satisfying update (Q_2, γ^k) at iteration k until convergence to (Q_2, γ^*).

To show the quality of registering two surfaces using Algorithm 3, we display an example of registering a cylinder S_2 (b) to an arbitrary surface S_1 (a) in Figure 1. The optimally registered surface S_2 as $(S_2, \hat{\gamma})$ is given in (c). We also provide three intermediate steps in (d) and the cost function $\|Q_1 - (Q_2, \hat{\gamma}^{iter})\|$ in (e) during iterations. For a better evaluation, we consider a linear interpolation α with $\alpha_0 = S_1$ (for $t = 0$) and $\alpha_1 = (S_2, \hat{\gamma})$ (for $t = 1$) in (f). In-between we show 4 uniform intermediate surfaces α_t for $5 * t = 1 \dots 4$. Note that the linear interpolation α is an extrinsic geodesic between $[Q_1]$ and $[Q_2]$ but may not coincide with a geodesic between them in $\mathcal{M}$. Yet, we observe that improved registration corresponds to smooth variations and better preservation of geometric features along α.

Algorithm 3 Successive Infinitesimal Variations

Input: Two Q-fields Q_1 and Q_2 with $||Q_1 - Q_2|| > 0$.
Output: The minimizer $\hat{\gamma}$.
1: Re-parametrize Q_1 and Q_2 uniformly.
2: Set $\hat{\gamma} = \mathrm{Id}_\Gamma$ and $Q = Q_2$.
3: Set $k = 1$ and $N \geq 3$.
4: **repeat**
5: Choose the step size $\rho = \frac{k}{N}$ where N represents maximum iterations.
6: Compute $Q^k = P\left(Q + \rho(Q_1 - Q)\right)$;
7: Compute γ^k from Q to Q^k using algorithm 2 ;
8: Update $Q = (Q, \gamma^k)$;
9: Update $\hat{\gamma} = \hat{\gamma} \circ \gamma^k$;
10: Update $k = k + 1$;
11: **until** $\rho = 1$ or $d(\gamma^k, \mathrm{Id}_\Gamma)$ very small

5. Relationship to Open Surfaces

Given two tubular surfaces S_1 and S_2, if a ξ_1-curve of S_1 fixedly matches to a ξ_1-curve of S_2, then the reparameterization problem between these two tubular surfaces can be viewed as a reparameterization problem between two open surfaces. An open surface is a differentiable function $S : [0,1]^2 \to \mathbb{R}^3 : [\xi_1, \xi_2] \mapsto S(\xi_1, \xi_2)$. Its normal field is defined as follows: $N : [0,1]^2 \to \mathbb{R}^3 : (\xi_1, \xi_2) \mapsto N(\xi_1, \xi_2) = \frac{\partial S}{\partial \xi_1} \times \frac{\partial S}{\partial \xi_2}$ and the corresponding Q-field is defined to be $Q(\xi_1, \xi_2) = \frac{N(\xi_1, \xi_2)}{\sqrt{\|N(\xi_1, \xi_2)\|_2}}$. It can be shown that the space of all Q-fields of open surfaces is the $\mathbb{L}^2$ space and the reparameterization group action on Q-fields is isometric [21].

We point out here without proofs that a connection between SRNF of open surfaces and SRVF of curves can be built similarly (see Lemma 5.1) and Algorithm 2 can be modified slightly for open surfaces (modify $\mathbb{S}^1$ to be $[0,1]$ in Lines 5 and 10 of Algorithm 2). Since the space of all Q-fields of open surfaces forms the $\mathbb{L}^2$ space, the geodesic between any two Q-fields is just the straight line between them, see details in Proposition 5.1. Note that the geodesics between any Q-fields of tubular surfaces are not the linear combinations since the space $\bar{\mathcal{M}}$ is a nonlinear subspace of $\mathbb{L}^2$.

Lemma 5.1. *Given two open surfaces S_1 and S_2 and their corresponding Q-fields Q_1 and Q_2. Let $\gamma = (\gamma_1, \gamma_2) \in \Gamma([0,1]^2)$. $\alpha_\lambda(\xi_2)$ denote $\gamma_2(\lambda, \xi_2)$.*

1. *Suppose $\gamma_1(\xi_1, \xi_2) = \xi_1$ for all $\xi_2 \in [0,1]$. Then for any given $\lambda \in [0,1]$, the function $h_\lambda(\alpha_\lambda) = \int_{\mathbb{S}^1} \left\| Q_1(\lambda, \xi_2) - (Q_2, \gamma)(\lambda, \xi_2) \right\|_2^2 d\xi_2$ can be reformulated into the form of* (3).
2. *Likewise, let $\beta_\lambda(\xi_1)$ denote $\gamma_1(\xi_1, \lambda)$. If $\gamma_2(\xi_1, \xi_2) = \xi_2$ holds for all $\xi_1 \in [0,1]$, the function $k_\lambda(\beta_\lambda) = \int_0^1 \left\| Q_1(\xi_1, \lambda) - (Q_2, \gamma)(\xi_1, \lambda) \right\|_2^2 d\xi_1$ also can be reformulated into the form of* (3).

Proposition 5.1. *Given two open surfaces S_0 and S_1, and an optimal parametrization $\hat{\gamma}$ between them, a geodesic between $[Q_0]$ and $[Q_1]$ is given by: $Q_t(\xi_1, \xi_2) = (1-t)Q_0(\xi_1, \xi_2) + t(Q_1, \hat{\gamma})(\xi_1, \xi_2)$.*

Proof for Proposition 5.1: According to (6), $\hat{\gamma}$ is a minimizer of

$$d^2([Q_0], [Q_1]) = \int_\Omega \|Q_0(\xi_1, \xi_2) - (Q_1, \hat{\gamma})(\xi_1, \xi_2)\|_2^2 d\xi_1 d\xi_2.$$

Remind that the space of Q-fields is a linear space, then the geodesic between Q_0 and $(Q_1, \hat{\gamma})$ is given by a straight line:

$$\alpha_t(\xi_1, \xi_2) = (1-t)Q_0(\xi_1, \xi_2) + t(Q_1, \hat{\gamma})(\xi_1, \xi_2)$$

minimizing the cost function

$$\|Q_0 - (Q_1, \hat{\gamma})\|_{\mathbb{L}^2} = \int_0^1 \sqrt{\left\langle \frac{d}{dt}\alpha_t, \frac{d}{dt}\alpha_t \right\rangle_{\mathbb{L}^2}} dt$$

with $\alpha_0 = Q_0$ and $\alpha_1 = (Q_1, \hat{\gamma})$. □

Note that for this special case the projection operator P used in Algorithm 3 in Line 6 is the identity. Except that, the same algorithm can be applied directly without modification to find an optimal parametrization $\hat{\gamma}$ between any two open surfaces S_1 and S_2.

6. Experimental Results

In this section, we present various examples using both synthetic surfaces and several benchmarks to show the accuracy of the proposed method. Since our framework extends registration of curves to surfaces, we present a comparison when using dynamic programming and a Riemannian optimization method to solve correspondence between curves. Usually parametrizations depend on the application at hand but one can still compute a parametrization of a given surface S as levelset of a differentiable scalar function [33]. Practical examples of functions computations over curved surfaces can be found in [34]. Similarly, all surfaces used in our experiments are represented by a finite set of ξ_1-curves and ξ_2-curves. Figure 2(a) gives an example of a parametrization (ξ_1, ξ_2) in (d) where ξ_1 is the parameter along axial axis (b) and ξ_2 is the parameter along circular axis (c). Since those curves are represented by 100 points each and therefore, the Q-field is also represented by points that are on some smooth function.

In all of the cost functions considered, Q_2 is composed with some function and, therefore, representing Q_2 (all curves) as a set of points is not sufficient and a suitable function must be used. Since the representation requires a C^1 function to compute partial derivatives, an interpolatory cubic spline on ξ_1 and ξ_2 is used.

It should be noted however that there is nothing in the formulation that requires an interpolatory approximation. Finally, all derivatives are calculated with the central difference scheme and all integrals required by the algorithms are approximated by the Composite Trapezoidal Rule.

6.1. Shape Registration

We show various examples in Figure 4 left. In each row, we present an example of registering a target S_2 in the left column (a) to a reference S_1 in the second column (b) using the proposed method. The reparametrized surface $(S_2, \hat{\gamma})$ is given in the right column (c). For a better evaluation of the qualitative improvement, results are provided as a linear combination between S_1 and $(S_2, \hat{\gamma})$. We note that the resulting registration provides a smooth deformation between surfaces that are "close" (first row) or where S_2 has a simplified geometry (second row). The other example, consider surfaces with more complicated geometries and the outcome reconfirms that the proposed method successfully provides smooth evolutions and better preservations of geometric features along the paths.

In order to evaluate the overall quality of our framework when using different methods to register curves, we performed a comparison between dynamic programming (DP) and a Riemannian optimization method (RO) [28]. We began by computing distances using both methods between 1000 random pairwise selections from 30 surfaces, see Figure 4(d) for some samples. Using those distances, we computed the realized gain $(\|Q_1 - Q_2\|_{\mathbb{L}^2} - \|Q_1 - (Q_2, \hat{\gamma})\|_{\mathbb{L}^2})$. We consider the gain as a distribution and report the results as

histograms in Figure 4(e): (dark blue) for RO and transparent for DP. We note that the proposed method provides good results for both methods with an advantage for RO. For example, a gain less than 10 has been recorded by 37% cases for RO and 50% for DP. In addition, a gain between 30 and 40 has been recorded approximatively by 18% for RO and by only 8% for DP. Furthermore, in very few cases DP reached a gain superior to 100 whereas the maximum gain realized by RO is 100.

6.2. 2D Shapes Clustering

Two public datasets are used in the experiments: the Flavia leaf dataset [35] and the MPEG-7 dataset [36]. The Flavia leaf dataset contains images of 1907 leaves from 32 species. Figure 5(a) shows an example leaf from each species. MPEG-7 contains 1400 images in 70 clusters each of which contains 20 shapes. Figure 5(b) shows an example shape from each cluster. The boundary curves of the shapes are extracted and 100 uniformly-spaced points were used to represent the shape.

The two public datasets were used to compare the performances of Riemannian optimization based method [28] and dynamic programming [16]. Indeed, we used k-medoids algorithm to perform clustering [37]. It reminds us that k-medoids is a standard and a widely used clustering algorithm. We applied the algorithm on the pre-computed distance matrix. Since k-mediod method may find a local minimizer, we repeated the clustering process 100 times and then take an average of their correctness. The correctness of k-medoids method is reported in Table 1. For completeness, we also provide the average computational time for pairwise distance from [28] . Note that the computational time is dominated by computing the pairwise distances between all shapes. The Riemannian optimization based methods are able to achieve higher correctness using less computational time when compared to the dynamic programming method.

6.3. 3D Shapes Classification

A public 3D objects benchmark [38] is used for the experimental evaluation of the proposed method. The original 763 models were classified into 19 classes. Figure 6 shows several examples from different classes. First, the boundary surfaces of 3D objects were aligned (rotation and translation) to a reference model and uniformly scaled. Then, 50 levelset curves is extracted to represent a surface where each curve is represented by 100 equidistant points. Only the first and the last levelsets were removed to make sure that the resulting representation is well defined. Finally, for all surfaces, the starting points have been chosen as an intersection between each curve and a fixed reference half-plane.

Shape classification has been studied extensively in the literature[39, 40, 41], to cite but a few examples. Nonetheless, whereas the main aim is to show the importance of one-to-one correspondance when comparing shapes, we propose to classify 3D shapes based on 3D geometric representations. For all classification experiments, we extract different features (HOG [42], mean curvature H_S, Gaussian curvature K_S [43] and 3D wavelet coefficients [44]) and use them separately to represent a surface. We remind that the goal is to use standard descriptors from literature to show the classification accuracy before and after re-parametrization. Thus, the choice of the most performing descriptors is out of the scope of this work. Descriptors' vector are given as input to train a Gaussian Processes Autoencoder (**GPAE**) [45] with 50 and 60 hidden layers, respectively. We select 75% of the dataset for training and use the rest for test. In order to remove the test bias, we run the methods 10 times. At each run, we randomly select the training set and use the remaining for test. We then average the performance over the 10 runs. To show the effectiveness of our approach, we show the classification accuracy without re-parametrization in Table 2 and after re-parametrization (using our method) in Table 3.

Furthermore, in order to objectively evaluate the proposed classification method we compare the GPAE with Support Vector Machine (SVM) classifier [46, 47]. Table 2 and Table 3 show that GPAE outperforms SVM in terms of classification accuracies except for HOG descriptor. Also, we can remark that using a good correspondence improves the accuracy for both GPAE and SVM classifiers. From all experiments, we can

note that the classification accuracy is better after re-parametrization regardless of any input (surface representation) or the classifier (GPAE or SVM). In particular, we observe that the best accuracies are achieved when using the mean curvature H_S as a feature. This result is well illustrated by confusion matrices in Figure 7 where the mean curvature is used to represent shapes and GPAE to classify them. Nevertheless, we should keep in mind that in both cases, we applied a Gaussian filtering, removed rigid transformation and normalized surfaces which certainly improves the accuracy even before re-parametrization.

7. Conclusion

We have presented a new Riemannian-based method for shape registration and classification. The advantage of using the proposed method is that it simplifies the optimization over the space of all re-parametrization by exploiting available frameworks for shape analysis of 3D curves. Beyond the use for finding optimal correspondences, we wanted to show how this framework could be directly applied for 2D and 3D shape registration, clustering or classification. We have tested our method on several benchmarks that demonstrate the effectiveness of the proposed methodology. We used standard descriptors to represent a surface and two different models for classification. From the experiments, we have showed that the classification accuracy is better after re-parametrization regardless of any descriptor or classifier.

Acknowledgement

Chafik Samir was partially supported by the Research Funds from CNRS-Prime funding. Wen Huang was partially supported by the Fundamental Research Funds for the Central University (NO. 20720190060).

References

[1] G. Allasia, R. Cavoretto, A. De Rossi, Hermite-birkhoff interpolation on scattered data on the sphere and other manifolds, Applied Mathematics and Computation 318 (2018) 35–50.

[2] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104 (2) (2013) 154–171.

[3] F. Mémoli, Gromov–wasserstein distances and the metric approach to object matching, Foundations of Computational Mathematics 11 (4) (2011) 417—-487.

[4] A. Zeng, S. Song, M. Niessner, M. Fisher, J. Xiao, T. Funkhouser, 3dmatch: Learning local geometric descriptors from rgb-d reconstructions, in: IEEE CVPR, 2017.

[5] Y. Bengio, A. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[6] I. Kaltenmark, B. Charlier, N. Charon, A general framework for curve and surface comparison and registration with oriented varifolds, in: IEEE CVPR, 2017.

[7] M. A. Viergever, J. B. A. Maintz, S. Klein, K. Murphy, M. Staring, J. P. W. Pluim, A survey of medical image registration, Medical Image Analysis 33 (2016) 140 – 144.

[8] X. Li, Y. Chang, Non-uniform interpolatory subdivision surface, Applied Mathematics and Computation, 324 (2018) 239–253.

[9] X. Pennec, Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements, Journal of Mathematical Imaging and Vision 25 (1) (2006) 127–154.

[10] E. M. Stokely, S. Y. Wu, Surface parametrization and curvature measurement of arbitrary 3-d objects: five practical methods, IEEE Trans. Pattern Anal. Mach. Intell. 14 (8) (1992) 833–840.

[11] L. Cosmo, M. Panine, A. Rampini, M. Ovsjanikov, M. M. Bronstein, E. Rodolà, Isospectralization, or how to hear shape, style, and correspondence, CoRR abs/1811.11465. arXiv:1811.11465.

[12] C. Raposo, J. P. Barreto, 3d registration of curves and surfaces using local differential information, CoRR abs/1804.00637. arXiv:1804.00637.

[13] M. Micheli, P. W. Michor, D. Mumford, Sobolev metrics on diffeomorphism groups and the derived geometry of spaces of submanifolds, Izvestiya: Mathematics 77 (3) (2013) 541.

[14] A. Navas, Groups of circle diffeomorphisms, Chicago Lectures in Mathematics.

[15] M. Bauer, M. Bruveris, P. W. Michor, Overview of the geometries of shape spaces and diffeomorphism groups, Journal of Mathematical Imaging and Vision 50.

[16] A. Srivastava, E. Klassen, S. H. Joshi, I. H. Jermyn, Shape analysis of elastic curves in Euclidean spaces., IEEE Trans. Pattern Anal. Mach. Intell.e 33 (7) (2011) 1415–1428.

[17] A review of chaos-based firefly algorithms: Perspectives and research challenges, Applied Mathematics and Computation 252 (2015) 155 – 165.

[18] W. Ring, B. Wirth, Optimization methods on riemannian manifolds and their application to shape space, SIAM Journal on Optimization 22 (2) (2012) 596–627.

[19] J. Glaunès, A. Trouvé;, L. Younes, Diffeomorphic matching of distributions: A new approach for unlabelled point-sets and sub-manifolds matching, IEEE CVPR 2 (2004) 712–718.

[20] M. Bauer, M. Bruveris, A New Riemannian Setting for Surface Registration, Proc. MICCAI Math. Foundations of Comp. Anatomy (2011) 182–193.

[21] I. H. Jermyn, S. Kurtek, E. Klassen, A. Srivastava, Elastic shape matching of parameterized surfaces using square root normal fields (2012) 804–817.

[22] S. Büyükkütük, İ. Kişi, G. Öztürk, Some Characterizations of Focal Surfaces of A Tubular Surface in $E\hat{3}$, arXiv e-prints (2018) arXiv:1810.05531arXiv:1810.05531.

[23] J. Schicho, Proper parametrization of real tubular surfaces, Journal of Symbolic Computation 30 (5) (2000) 583 – 593.

[24] S. Hu, Z. Wang, X. Tang, Tubular surfaces of center curves on spacelike surfaces in lorentz-minkowski 3-space, Mathematical Methods in the Applied Sciences 42 (9) (2019) 3136–3166.

[25] V. Mohan, G. Sundaramoorthi, A. Tannenbaum, Tubular surface segmentation for extracting anatomical structures from medical imagery, IEEE transactions on medical imaging 29 (12) (2010) 1945–1958.

[26] H. Zhou, J. K. Min, G. Xiong, Implicit Tubular Surface Generation Guided by Centerline, arXiv e-prints (2016) arXiv:1606.03014.

[27] T. Maekawa, N. M. Patrikalakis, T. Sakkalis, G. Yu, Analysis and applications of pipe surfaces, Computer Aided Geometric Design 15 (1998) 437–458.

[28] W. Huang, K. A. Gallivan, A. Srivastava, P.-A. Absil, Riemannian optimization for registration of curves in elastic shape analysis, Journal of Mathematical Imaging and Vision 54 (3) (2016) 320–343.

[29] S. Kurtek, E. Klassen, Z. Ding, A. Srivastava, A novel Riemannian framework for shape analysis of 3D objects, IEEE CVPR (2010) 1625–1632.

[30] U. Grenander, M. I. Miller, Pattern Theory: From Representation to Inference, Oxford University Press, 2007.

[31] L. Younes, F. Arrate, M. I. Miller, Evolutions equations in computational anatomy, NeuroImage 45 (1) (2009) 40–50.

[32] J. A. Carrillo, J. S. Moll, Numerical simulation of diffusive and aggregation phenomena in nonlinear continuity equations by evolving diffeomorphisms, SIAM Journal on Scientific Computing 31 (6) (2010) 4305–4329.

[33] S. Osher, R. P. Fedkiw, The Level Set Methods and Dynamic Implicit Surfaces, Vol. 57, 2004.

[34] O. Nilsson, Level-set methods and geodesic distance functions, Thesis, LinkOping University, 2009.

[35] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, Q.-L. Xiang, A leaf recognition algorithm for plant classification using probabilistic neural network, IEEE ISSPIT (2007) 11–16.

[36] D. of Computer Information, T. U. Science, Shape similarity research project, www.dabi.temple.edu/ shape/MPEG7/dataset.html.

[37] L. Kaufman, P. J. Rousseeuw, Clustering by means of medoids (1987).

[38] A. Koutsoudis, G. Pavlidis, V. Liami, D. Tsiafakis, C. Chamzas, 3d pottery content-based retrieval based on pose normalisation and segmentation, Journal of Cultural Heritage 11 (3) (2010) 329 – 338.

[39] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, J. Xiao, 3d shapenets: A deep representation for volumetric shapes, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1912–1920.

[40] J.-X. Du, X.-F. Wang, G.-J. Zhang, Leaf shape based plant species recognition, Applied Mathematics and Computation 185 (2) (2007) 883 – 893.

[41] S. Biasotti, B. Falcidieno, D. Giorgi, M. Spagnuolo, Mathematical tools for shape analysis and description, Synthesis Lectures on Computer Graphics and Animation 6 (2) (2014) 1–138.

[42] T. Vo, D. Tran, W. Ma, K. Nguyen, Improved HOG descriptors in image classification with CP decomposition, in: Neural Information Processing, 2013, pp. 384–391.

[43] B. Levy, Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry, in: IEEE International Conference on Shape Modeling and Applications 2006, 2006, pp. 13–13.

[44] Y. D. Cid, H. Müller, A. Platon, P. Poletti, A. Depeursinge, 3d solid texture classification using locally-oriented wavelet transforms, IEEE Trans. Image Processing 26 (4) (2017) 1899–1910.

[45] J. Snoek, R. P. Adams, H. Larochelle, Nonparametric guidance of autoencoder representations using label information, J. Mach. Learn. Res. 13 (1) (2012) 2567–2588.

[46] M. Caputo, K. Denker, M. O. Franz, P. Laube, G. Umlauf, Support vector machines for classification of geometric primitives in point clouds, in: Curves and Surfaces, Springer International Publishing, 2015, pp. 80–95.

[47] Yi Liu, Y. F. Zheng, One-against-all multi-class svm classification using reliability measures, in: IEEE International Joint Conference on Neural Networks, Vol. 2, 2005, pp. 849–854 vol. 2.

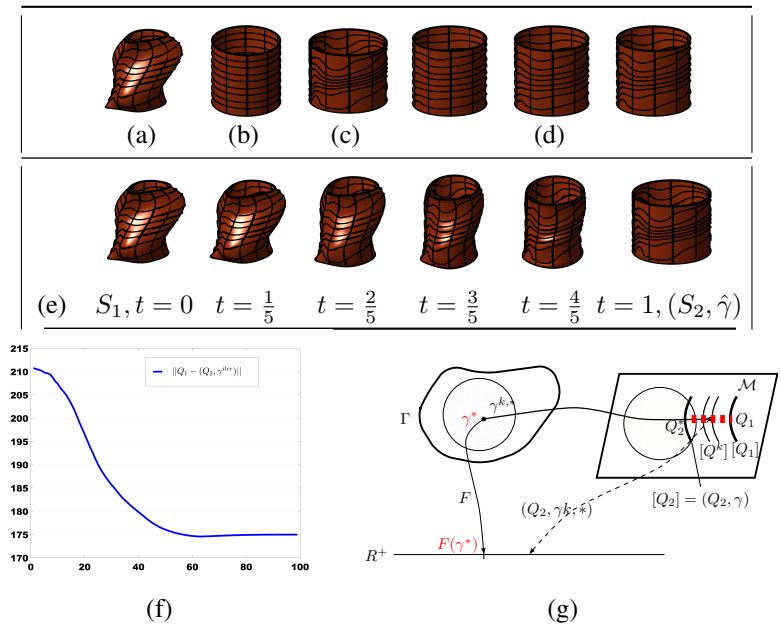

Figure 1: Reparameterization of the target surface S_2 (b) to match the reference surface S_1 (a) using the proposed method. (c) The optimally registered target as $(S_2, \hat{\gamma})$. (d) Examples of intermediate steps and (e) the linear combination path between S_1 and $(S_2, \hat{\gamma})$. (f) The cost function $\|Q_1 - (Q_2, \hat{\gamma}^{iter})\|_{\mathbb{L}^2}$ during iterations and (g) an illustration of the optimization process initialized at $Q_2^0 = (Q_2, Id_\Gamma)$ and ending at the optimal solution $Q_2^* = (Q_2, \hat{\gamma})$.

Figure 2: Example of a tubular surface S (a) and the corresponding, ξ_1-curves (axial), ξ_2-curves (circular), and parametrization (ξ_1, ξ_2).

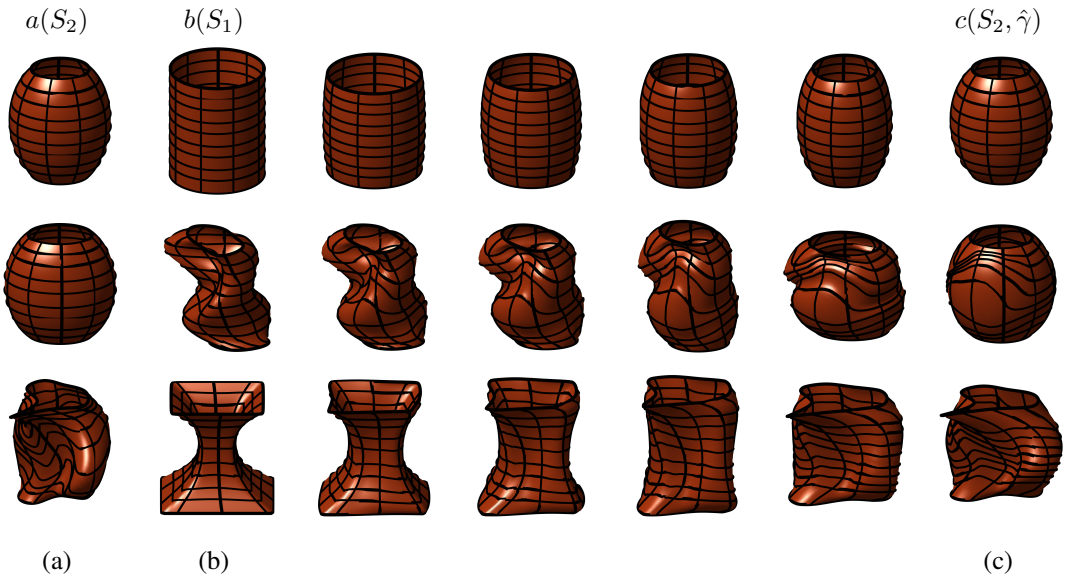

Figure 3: Reparameterization of a surface S_2 (b) as one-to-one correspondence with S_1 (a) using the proposed method. (c) The optimally registered target as $(S_2, \hat{\gamma})$. For each row, the linear combination path between S_1 and $(S_2, \hat{\gamma})$ is given from second column (b) to the right column (c).

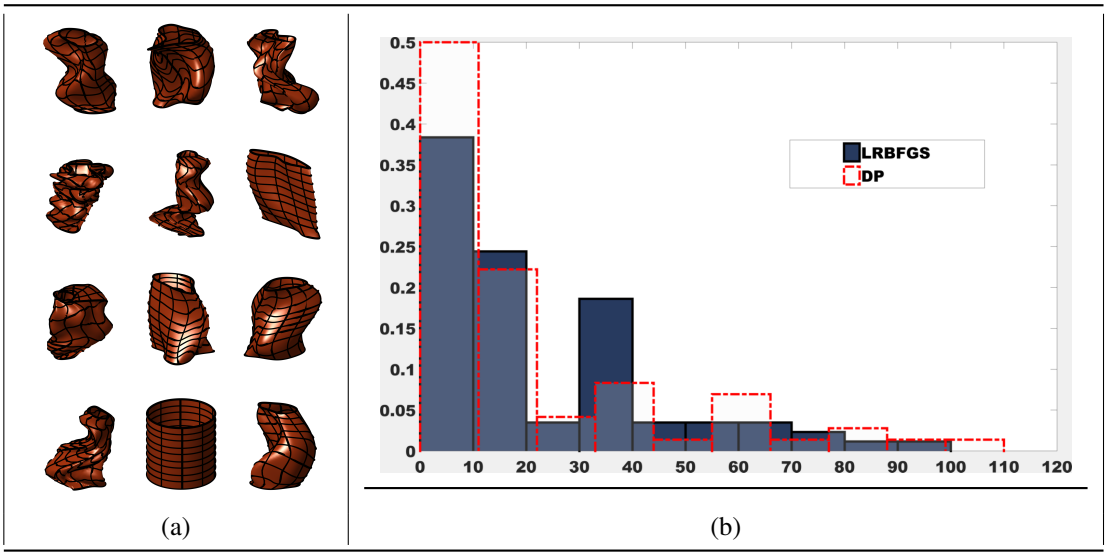

Figure 4: (a) Samples of surfaces used for experiments and (b) histograms of the realized gain between the starting distance before re-parametrization $\|Q_1 - Q_2\|_{\mathbb{L}^2}$ and the distance after re-parametrization$\|Q_1 - (Q_2, \hat{\gamma})\|_{\mathbb{L}^2}$ using the Riemannian method (dark blue) and DP (red dashed).

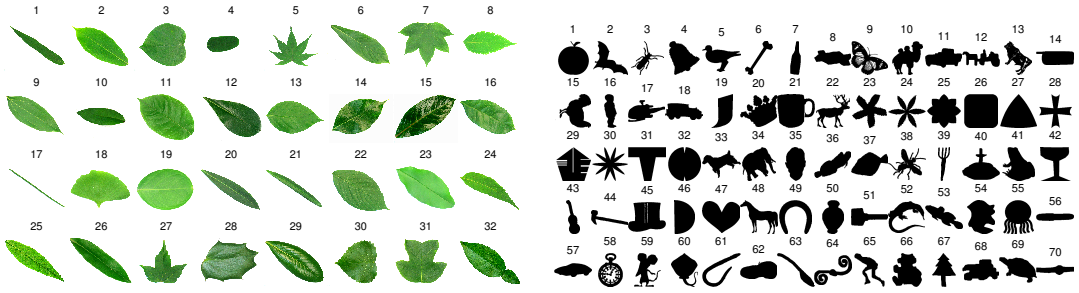

Figure 5: (a): Samples of leaves from the Flavia leaf dataset. (b) Samples of curves from the MPEG-7 dataset. One sample per cluster is illustrated.

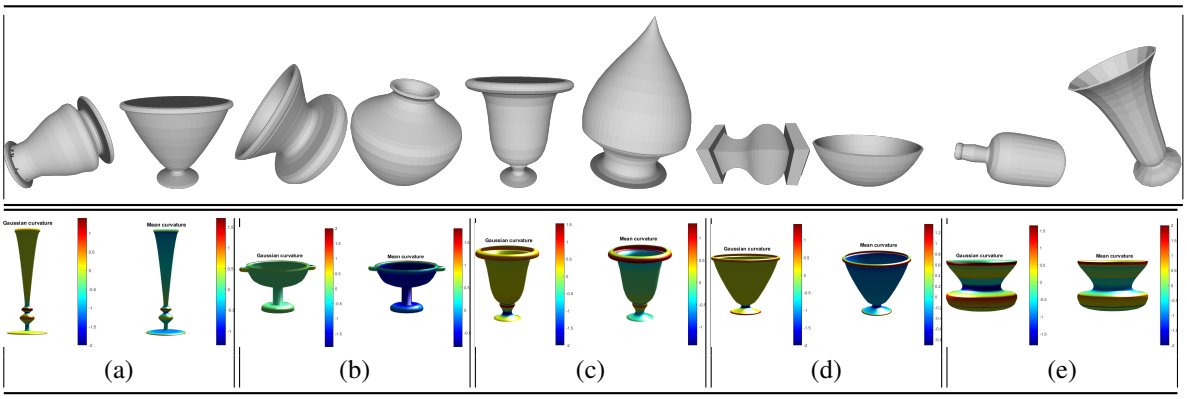

Figure 6: Top: Samples of 3D models from the 3D dataset. Bottom: Examples after rigid alignment and re-parametrization colored with curvature extraction (for each column a-e Gaussian left and mean right).

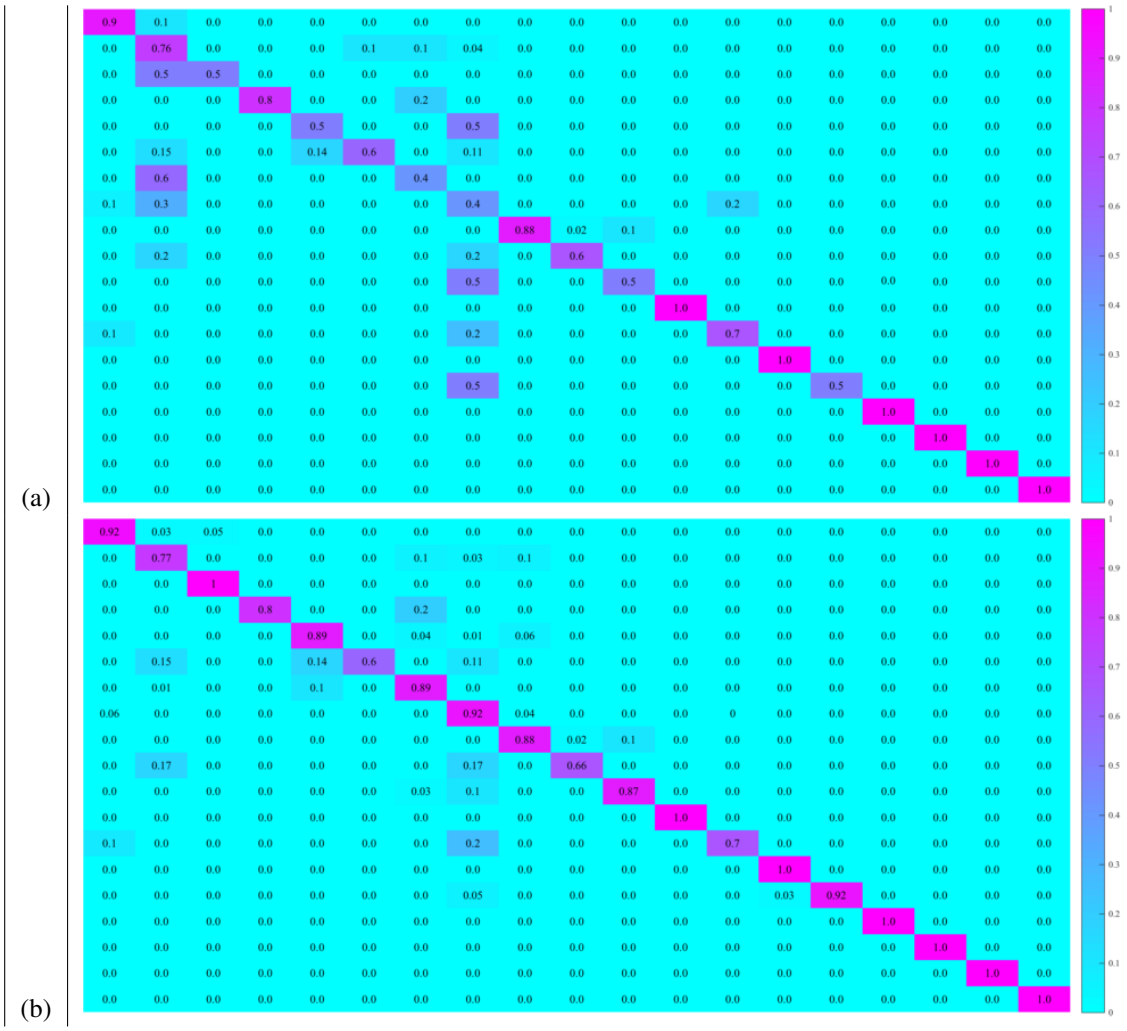

Figure 7: Confusion matrices using the mean curvature with GPAE: (a) before and (b) after re-parametrization.

Table 1: 2D shapes classification rate. Two versions of limited-memory Riemannian BFGS (RO) based methods are used.

	RO-complex	RO-simple	DP
FLAVIA (rate)	68,94%	68.37%	66.53%
FLAVIA (time)	0.088	0.047	0.897
MPEG-7 (rate)	68,94%	68.37%	66.53%
MPEG-7 (time)	0.181	0.134	0.908

Table 2: 3D shapes classification accuracy before re-parametrization.

Method	Results	HOG	Wavelet	H_S	K_S
GPAE	rate	73.1%	**78**%	76.8%	73.9%
	σ	1.1	0.8	1.7	2.0
SVM	rate	74.2%	74.5%	**75.6**%	70.2%
	σ	1.0	1.2	0.9	1.5

Table 3: 3D shapes classification accuracy after re-parametrization.

Method	Results	HOG	Wavelet	H_S	K_S
GPAE	rate	77.8%	79.%	**87.5**%	80.1%
	σ	1.5	1.4	1.8	1.9
SVM	rate	**79.4**%	75.0%	77.3%	78.2%
	σ	1.1	1.0	1.6	2.0