J Math Imaging Vis (2010) 38: 226–240
DOI 10.1007/s10851-010-0218-2

On Restricting Planar Curve Evolution to Finite Dimensional Implicit Subspaces with Non-Euclidean Metric

Aditya Tatu · François Lauze · Stefan Sommer · Mads Nielsen

Published online: 14 August 2010

Abstract This paper deals with restricting curve evolution to a finite and not necessarily flat space of curves, obtained as a subspace of the infinite dimensional space of planar curves endowed with the usual but weak parametrization invariant curve L^2-metric.

We first show how to solve differential equations on a finite dimensional Riemannian manifold defined implicitly as a submanifold of a parameterized one, which in turn may be a Riemannian submanifold of an infinite dimensional one, using some optimal control techniques.

We give an elementary example of the technique on a spherical submanifold of a 3-sphere and then a series of examples on a highly non-linear subspace of the space of closed spline curves, where we have restricted mean curvature motion, Geodesic Active contours and compute geodesic between two curves.

Keywords Curve evolution · Active contours · Riemannian geometry · Exponential map · Log map · Shooting method · Shape space · Constrained optimization · Nonlinear projection · Implicitly defined shape space

A. Tatu (✉) · F. Lauze · S. Sommer · M. Nielsen
Institute of Computer Science, University of Copenhagen, Copenhagen, Denmark
e-mail: aditya@diku.dk

F. Lauze
e-mail: francois@diku.dk

M. Nielsen
e-mail: madsn@diku.dk

1 Introduction

Curve evolution[1] has now become a standard tool in Computer Vision. It has been used for tracking interfaces, registration, active contour algorithms for segmentation, etc. [1–4]. Curve evolution involves deformation or motion of a curve via a given velocity vector field defined on a space of curves that share some common structures. Curves can be manipulated implicitly as the zero-level set of a given function for instance, or explicitly via a parametrization. In this paper we are interested in the latter. We work on the space of smooth parametrized curves, which is a Riemannian manifold.

The general curve evolution equation is given as,

$$\frac{\partial c}{\partial t}(p,t) = v(p,t)\vec{n}(p,t) + w(p,t)\vec{t}(p,t) \tag{1}$$

where $c(-,-)$ is the family of curves satisfying the equation, p is the parameter along a curve, t is the time parameter, v is the scalar velocity in the unit normal direction $\vec{n}$ of the curve, w is the scalar velocity in the tangential direction $\vec{t}$ to the curve. In [5], the authors show that the tangential component of the velocity affects only the parametrization of the curve, thus can be omitted and we get

$$\frac{\partial c}{\partial t}(p,t) = v(p,t)\vec{n}(p,t). \tag{2}$$

These equations arise either as a 'direct design' approach where the velocity vector is given according to some application specific requirement, or as a gradient descent scheme in order to minimize an energy functional $E : \mathcal{S} \to \mathbb{R}$, where $\mathcal{S}$ is the space of curves in which we are interested.

[1] We deal with planar curves only.

Only relatively recently emphasis was set on the fact that the gradient of the functional depends on the definition of the inner product on $\mathcal{S}$, as noticed by [6, 7] making this kind of flow fundamentally dependent of the metric. Even when equations arise from a direct design and do not involve the notion of gradient, their numerical implementation may require understanding of the metric, as one may leave the manifold due to small, but *non infinitesimal* steps.

Probably the most used inner product in applications is the standard reparametrization invariant L_2 inner product between *normal* curve deformations v_1 and v_2 of a curve $c(p)$:

$$\langle v_1, v_2\rangle_c = \int_0^N v_1(p)v_2(p)|c'(p)|\,dp. \tag{3}$$

As it will play a fundamental role in the sequel, we denote it as $\langle -, -\rangle_c$, where c is the point, i.e. a curve, at which it is computed. When there is no ambiguity, we may omit the subscript c. This inner product was used for the Geodesic Active Contours of Caselles et al. [8].

Sundaramoorthi et al. in [9], as well as Charpiat et al. [10] have used Sobolev inner products for active contour segmentation, thereby avoiding local minima amongst other advantages.

Curve evolution is also implicitly used in applications like shape matching and classification based on shapes, since it involves deforming or evolving one shape into another. The space of curves $\mathcal{S}$ under consideration may vary from application to application, but will not necessarily be linear and flat, and non flatness makes it an ideal ground for use of differential geometric tools.

Michor and Mumford [11] work on the space of smooth embeddings $Emb(\mathbb{S}^1, \mathbb{R}^2)$ and smooth immersions $Imm(\mathbb{S}^1, \mathbb{R}^2)$. The difference between the two is that smooth embeddings are such that the closed curves obtained by embedding the circle into the plane, should have the same topology as $\mathbb{S}^1$, while immersed one can have self intersections. When identifying reparametrization, the two spaces above give rise to two spaces of curves:

$$B_e = Emb(\mathbb{S}^1, \mathbb{R}^2)/Diff(\mathbb{S}^1)$$

and

$$B_i = Imm(\mathbb{S}^1, \mathbb{R}^2)/Diff(\mathbb{S}^1)$$

(note $B_e \subset B_i$). In [11], the authors prove that the L^2 metric is a weak Riemannian metric: it degenerates on B_e and B_i, i.e. the geodesic distance between any two curves is zero. Given a curve, one can perturb it using infinitesimal deformation with very high frequencies towards any other given curve. These deformations are infinitely close to any curve and can be used to build arbitrarily short paths between any two curves, i.e. the corresponding geodesic distance degenerates. This implies that one cannot compute distance between shapes using this inner product. This of course does not prevent other algorithms to work fine, this is the case for the above mentioned Geodesic Active Contours, although even in that case, the weak structure is the cause of potential spurious local minima.

There have been several strategies to overcome this hurdle. One is to change the metric on the space of curves, for e.g. in [9], the authors use a Sobolev metric and in [11] the authors use a curvature dependent metric, both of which restrict the spatial variation in the deformations. In [12], authors use a Sobolev-type metric that is also tailored for similarity transform, similar to [10]. Glaunes et al. [13] represent curves as a linear functional on vector fields over $\mathbb{R}^d$ and compare curves by proposing norms over the dual space of vector fields on $\mathbb{R}^d$. These norms depend on spatial derivatives, which limit the variation of the vector fields, prohibiting high frequency variations. In [14, 15], a group theoretic approach is used. The set of admissible deformations forms a group and then the problem to compute geodesic distance between two given curves is reduced to computing the geodesic distance between two members of the group of deformations that map one curve into another. They also consider spatial derivatives on infinitesimal deformations for constructing the inner product, thereby limiting the spatial variations. Also using a group theoretic approach, Micheli constructs in [16] manifolds of landmark shapes in which landmark geodesic curves are obtained via coupling with diffeomorphisms. Related to landmark based shapes are the polygonal shapes used by Unal et al. in [17], but no Riemannian structure is directly used.

Landmark manifolds are finite dimensional and the associated Riemannian metric cannot be weak. The same will hold on a finite subspace of the infinite dimensional space of all curves. The space of closed curves with finite Fourier expansion is used in [18] to represent curve deformations. There is an upper bound on the frequency of deformations that one can introduce in the finite dimensional case and hence the metric does not degenerate on this subspace. The finite dimensional case is also important since even though theoretically the intended curve evolution happens on the infinite dimensional space of all curves, but when implemented on a computer, it is restricted to some finite dimensional subspace. In [19], the authors restrict curve evolution to linear finite dimensional subspace of curves. The present work is an extension of this one, using a finite dimensional subspace of B_i. Due to both the structure of B_i and the type of restrictions we may impose, this subspace will often be curved. We work on problems where such specifications, i.e. finite dimensional curve spaces are given implicitly.[2]

[2]Explicit constraints give a parametrization of the subspace, which can be used to compute standard differential geometric maps like the exponential map, and thus solve the problem.

The problem (of restricting curve evolution) corresponds to solving an ordinary differential equation on a manifold $\mathcal{S}$, of the form

$$\frac{dc(t)}{dt} = F(c(t)), \quad F(c) \in T_c\mathcal{S} \tag{4}$$

where $T_c\mathcal{S}$ is the tangent space of $\mathcal{S}$ at c. In numerical problem $\mathcal{S}$ is embedded into a larger space $\mathcal{T}$ such that $T_c\mathcal{S}$ can naturally be represented, at least locally as a subspace of $\mathcal{T}$, while $\mathcal{T}$ is normally itself an open of a numeric space $\mathbb{R}^n$. The forward Euler step

$$c + \delta t \frac{dc}{dt}$$

will generally leave $\mathcal{S}$ and we need a way to project back to it. Hairer [20] investigates several projection methods to get back to the manifold. Srivastava et al. [18] use a gradient descent approach to project curve deformations from the tangent space of the given subspace of curves, to curves in the subspace.

In our work, we directly exploit *the L^2-metric of the space of immersed closed curves* to perform the natural projection, given by the Riemannian exponential map, it computes the target *geodesic* path leaving from c with initial velocity $\delta t dc/dt$ and of length exactly $\|\delta t dc/dt\|$. We derive an formulation of it for the case where $\mathcal{T}$ is not endowed the usual flat metric. From this formulation, one obtains by proper differentiations, a system of differential equations that computes parallel transport. This is the main ingredient of the shooting method used to compute the Riemannian Log map, which allows to compute geodesic between two given points, in our case minimal deformations from curve c to curve d.

We apply it to the subspace of closed B-spline curves consisting of curves with equidistant neighboring node (images of knots in the curve), with the L^2 metric induced from B_i.

This paper is organized as follows. In Sect. 2 we state and describe the projected evolution problem and describe our algorithm from a general point of view, with particular emphasis on computation of Riemannian Exponential maps, Log maps and parallel transports. We also give a toy example demonstrating our method towards the end of the same section. In Sect. 3, we describe a particular subspace of spline curves that we are interested in, and derive the necessary formulae. In Sect. 4, we give some results of restricting the curvature flow and geodesic active contours to the subspace described in Sect. 3. We also demonstrate the computation of a geodesic between two curves and we conclude in Sect. 5. For the reader's convenience, we have collected basic facts from Riemannian geometry in Appendix.

2 Projected Curve Evolution

We consider the following situation. A Riemannian manifold $\mathcal{T}$ possibly infinite dimensional is given with a submanifold $\mathcal{S}$ of dimension N, that is described via a chart $g : \mathcal{S} \to U \subset \mathbb{R}^N$, and a submanifold V of $\mathcal{S}$ defined as $F^{-1}(0)$ where $F : \mathcal{S} \to \mathbb{R}^m$ is a submersion (we could also work with its representation in parameters space $\tilde{F} : U \to \mathbb{R}^m$). We have of course in mind the case where $\mathcal{S}$ is a finite dimensional submanifold of $\mathcal{T} := B_i$ with the induced metric. The submersion condition means that the Jacobian $JF(x)$ of F is onto at every $x \in V$, this is this condition that makes V a smooth submanifold of $\mathcal{S}$ (see [21]). V is then endowed with the metric induced by $\mathcal{S}$ to become a Riemannian submanifold of $\mathcal{S}$.

In Sect. 3, we consider a slightly more complicated hierarchy with the space of spline curves, but we defer the discussion to that particular section.

Then, given an initial point $c \in V$ and an evolution equation of the form

$$\frac{dc}{dt} = \mathcal{F}(c), \quad \mathcal{F}(c) \in T_c\mathcal{T} \tag{5}$$

we require a solution $c(t)$, such that $c(t) \in V$ for $t \geq 0$. Obviously the velocity vector may evolve a given point out of V, hence we need to modify (5) by introducing some projection steps to ensure that the solution stays in V. In brief, one iteration of our algorithm consists of the following three steps:

1. Compute $v_S = \Pi_{T_{c(t)}\mathcal{S}}\mathcal{F}(c)$, the orthogonal projection of velocity to the tangent space of $\mathcal{S}$ at $c(t)$.
2. Compute $v_V = \Pi_{T_{c(t)}V} v_S$, the orthogonal projection to the tangent space of the implicitly defined subspace V.
3. Compute $c(t + dt) = Exp^V_{c(t)}(v_V\, dt)$, the V-exponential map[3] to map the velocity vector from the tangent space of V to the subspace V.

In our main application, $\mathcal{S}$ will be a submanifold of B_i and its metric will be induced by the reparametrization invariant L^2-metric (3). That is, even though via the local coordinate system g, elements of $\mathcal{S}$ can be seen as points of $\mathbb{R}^N$, they carry a non flat metric. Since we are now dealing with *finite dimensional manifolds*, *we will not encounter the geodesic distance degeneracy proved in* [11] for the "standard" inner product defined in (3) as the corresponding geodesic distances on $\mathcal{S}$ and V must give back the topologies of these spaces (see [21]). This can be explained by the fact that varying one parameter of the parametric representation of the curve induces a non-local variation of the curve with curvature change bounded with parameter change.

[3] We will drop the superscript V from Exp^V in the sequel when there is no ambiguity.

2.1 Orthogonal Projections

The two projections of step 1 and 2 above are of the same nature and are local versions of the classical orthogonal projection on a finite dimensional subspace of a Hilbert space [22]: given a Riemannian manifold M, a *finite dimensional* submanifold N and $n \in N \subset M$, one has $T_nN \subset T_nM$ and thanks to the Riemannian structure, an orthogonal projection Π_{TN}^{TM} at n that can be computed as follows. Let $\mathbf{v} \in T_mM$ and $\{e_1, \ldots, e_k\}$ a basis, not necessarily orthogonal of T_nN. Then $\mathbf{v}_N = \sum_{i=1}^{k} \alpha_i e_i$ is the projection of $\mathbf{v}$ if and only if $\mathbf{v} - \mathbf{v}_N$ is orthogonal to T_nN as a subspace of T_nM i.e.

$$\langle \mathbf{v} - \mathbf{v}_N, e_i \rangle_n = 0 \quad \forall i = 1, \ldots, k$$

($\langle \cdot, \cdot \rangle_n$ is the inner product in T_nM that comes from the Riemannian structure). Set $G := (\langle e_i, e_j \rangle_n)_{ij}$, this is the *Gram matrix* of $e = (e_1, \ldots, e_k)$. Then the projected velocity is given by

$$\mathbf{v}_N = \Pi_{TN}^{TM}(\mathbf{v}) = \sum_i [G^{-1}(\langle \mathbf{v}, e \rangle_n)]_i \; e_i. \tag{6}$$

Given $c \in V \subset \mathcal{S} \subset \mathcal{T}$, we apply this first to $c \in \mathcal{S} \subset \mathcal{T}$, as $\mathcal{S}$ has finite dimension, to get the projection $\Pi_{T_{c(t)}\mathcal{S}}$ of step 1. The tangent space T_cV is the kernel (or null-space) of $DF(c)$ and also of finite dimension and we apply (6) to this configuration and get $\mathbf{v}_V = \Pi_{T_{c(t)}V}$.

2.2 Exponential Map on the Submanifold V

Even if T_cV can be seen as a subspace of $\mathcal{S}$, a standard Eulerian step

$$c(t + dt) = c(t) + \mathbf{v}_V \, dt$$

where dt is a finite time step will result in $c(t + dt) \notin V$ generally. We use the exponential map to project the tangent vector $\mathbf{v}_V \, dt$ to V, i.e.,

$$c(t + dt) = Exp_{c(t)}(\mathbf{v}_V \, dt).$$

Computing the Exponential map on implicitly defined manifolds is not straightforward as in the case when manifolds are given via local parametrization (such a computation boils down to solving a classical second order ODE). We obtain instead the Exponential map via Optimal control theory [23, 24], along the lines of Dedieu and Nowicki [25]. In their setting V is a submanifold of $\mathbb{R}^N$, the latter endowed with its usual Euclidean inner product whereas in our case, we consider more complex structures on $\mathcal{S}$.

We assumed in the introduction of this section that we had a local coordinate system $g : \mathcal{S} \to U \subset \mathbb{R}^N$, through which we now assume $\mathcal{S}$ to be an open subset of $\mathbb{R}^N$. Its Riemannian structure is given by the smooth family of metrics

$$x \in U \mapsto G(x)$$

with $G(x)$ a symmetric and positive definite $N \times N$ matrix.

A curve $x(t), t \in [0, 1]$ is a geodesic on V if it minimizes the constraint length

$$\ell(x) = \int_0^1 \|\dot{x}(t)\|_{G(x(t))} \, dt, \quad x(t) \in V, \; \forall t \in [0, 1]$$

where $\dot{x}(t)$ is the velocity of x at time t, $\|\dot{x}\|_G(x) = \sqrt{\dot{x}^T G(x) \dot{x}}$ is the length of vector $\dot{x}$ (we will often omit from now the parameter t). Such a curve is equivalently a minimizer of the constrained curve energy [26]

$$\mathcal{E}(x) = \frac{1}{2} \int_0^1 \|\dot{x}\|^2_{G(x)} \, dt, \quad F(x(t)) = 0.$$

Introducing the constraint via the Lagrange multiplier $\lambda := \lambda(t) \in \mathbb{R}^m$, we have now the Lagrangian of the system

$$\mathcal{L}(x, \dot{x}, \lambda) = \frac{1}{2} \dot{x}^T G(x) \dot{x} - \lambda \cdot F(x)$$

which provides the second-order Euler-Lagrange equation

$$\frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{\partial \mathcal{L}}{\partial x} = 0. \tag{7}$$

Instead of solving this equation, we use the Hamiltonian formulation of [25] and define our Hamiltonian to be

$$H(p, x, \mu) = -\frac{1}{2} \dot{x}^T G(x) \dot{x} + p^T \dot{x} + \sum_{i=1}^{m} \mu_i J F_i(x) \dot{x} \tag{8}$$

where $p = p(t)$ is an auxiliary variable, $JF(x)$ is the Jacobian of F, $\mu = (\mu_1(t), \ldots, \mu_m(r))$ is a Lagrange multiplier. The first term in the Hamiltonian is the cost function (square of velocity length in this case) to be minimized and the last term is the constraint stating that the path should remain in V, i.e. $dF_i(x, \dot{x}) = \nabla F_i(x)^T \dot{x} = JF_i(x)\dot{x} = 0, \forall i$.

Pontryagin's maximum principle gives the necessary conditions for a maximum:

$$\frac{\partial H}{\partial \dot{x}} = 0, \tag{9}$$

$$\dot{p} = -\frac{\partial H}{\partial x}. \tag{10}$$

Compared to (7), this is a system of first-order equations, that can be solved by standard to more advanced ODE solvers. In particular, as it comes from a Hamiltonian, accurate symplectic solvers can be used.

Equation (9) in our case gives

$$G(x)^{-1} p = \dot{x} - J_G F(x)^T \mu$$

where $J_G F(x)$ is the Jacobian of F with respect to the metric $G(x)$ and is given as $J_G F(x) = JF(x)G(x)^{-1}$ (i.e., it is a matrix containing transpose of the gradient vector of the individual functions F_i, with respect to the metric G).

We know that $\dot{x} \in T_x V$ and $J_G F(x)^T \mu \in N_x V$, the normal space of V at x. Therefore, we have

$$\dot{x} = \Pi^G_{T_x V}(G(x)^{-1} p) \tag{11}$$

the orthogonal projection to the tangent space with respect to the inner product $G(x)$, and

$$-J_G F(x)^T \mu = \Pi^G_{N_x V}(G(x)^{-1} p),$$
$$\mu = -(J_G F(x)^T)^{\dagger} \Pi^G_{N_x V}(G(x)^{-1} p) \tag{12}$$

where $(J_G F(x)^T)^{\dagger}$ is the pseudo-inverse of $J_G F(x)^T$ *for the metric* G [25, 27], we call it G-pseudo-inverse in the sequel, and we show how to compute it at the end of this section. From (10), we get

$$\dot{p} = \frac{1}{2}\dot{x}^T DG_x(x)\dot{x} - \sum_{i=1}^{m} \mu_i HF_i(x)\dot{x} \tag{13}$$

where $DG_x(x)$ is the derivative of $G(x)$ with respect to x and $HF_i(x)$ is the Hessian of F_i at the point x. Note that DG and the HF_i are directly related to the curvature of $\mathcal{S}$ and the extrinsic curvature (second fundamental form) of V in $\mathbb{S}$.

Now, given the initial conditions:

$$\begin{cases} \dot{x}(0) = u \\ \mu(0) = 0 \\ p(0) = G(x(0))u \end{cases}$$

one computes the Exponential map using finite difference approximations of (11), (12) and (13).

The G-pseudo-inverse in (12) is computed as follows. Let us denote $J_G F(x)^T$ by A. Since A^T is onto, A is injective. For an injective matrix A, the pseudo-inverse is given as,

$$A^{\dagger} = (A^* A)^{-1} A^*$$

where A^* is the adjoint operator of A. By definition of the adjoint operator, we have for $x \in \mathbb{R}^m$ (with usual inner product) and $y \in \mathbb{R}^N$ (with inner product given by G, $\langle x, y\rangle_G = x^t G y$),

$$\langle Ax, y\rangle_G = \langle x, A^* y\rangle_{\mathbb{R}^m}$$

which gives

$$A^* = A^t G.$$

So,

$$(J_G F(x)^T)^* = J_G F(x) G(x) = JF(x)$$

and the pseudo-inverse is given as

$$(J_G F(x)^T)^{\dagger} = (JF(x)G(x)^{-1} JF(x)^T)^{-1} JF(x). \tag{14}$$

These three projection steps are repeated for every iteration of the curve evolution.

2.3 Log Map Using Parallel Transport

Although Log map may not be required for evolving curves, it is necessary for computing geodesics between two given shapes.

While exponential maps are defined by initial value problems (IVP), the computation of geodesic between two points is a Two-Points boundary value problem (TPBVP) and shooting methods [28, 29] have been developed to solve these problems by transforming TPBVPs into IVPs, by an iterative procedure that refines estimates of initial values so that the resulting IVP solution will reach the boundary value.

This is what we will use here in order to compute the Log map between points in our implicitly defined space equipped with a non-Euclidean metric. The procedure consists of updating an initially chosen tangential direction with a vector obtained by parallel transporting an approximate error vector between the current point (obtained by the Exp map of the tangent direction in the current iterate) and the target point. A typical step is illustrated in Fig. 1 where the geodesic in V joining points q and $\tilde{q}$ is sought.

We describe the step in greater details. First the notations used in the figure. To avoid ambiguities, let $Exp^{\mathcal{S}}$ denote the exponential map in $\mathcal{S}$ and Exp^V the exponential in submanifold V. Similar notations $Log^{\mathcal{S}}$ and Log^V are used for the Riemannian Log maps. A geodesic joining points p and p' in $\mathcal{S}$ is denoted by $\gamma_{\mathcal{S}}(p, p')$.

Assume then that $\mathbf{v}^n$ is the current estimate of the needed initial velocity to reach $\tilde{q}$ in V and $q^n = Exp^V_q \mathbf{v}^n$ its exponential. If $q^n \neq \tilde{q}$ we want to build an update of the initial velocity $\mathbf{v}^n$ that takes into account the estimation error between q^n and $\tilde{q}$. We approximate this error in several steps. First we consider the geodesic $\gamma_{\mathcal{S}}(q^n, \tilde{q})$ in $\mathcal{S}$ linking q^n and $\tilde{q}$ and the log map $\mathbf{u}^n = Log^{\mathcal{S}}_{q^n} \tilde{q}$. If the extrinsic curvature of V in $\mathcal{S}$ is not too "large", the orthogonal projection of $\mathbf{u}^n$ on $T_{q^n} V$ provides an approximation of the V-Log map $Log^V_{q^n} \tilde{q}$. This however requires the computation of $\gamma_{\mathcal{S}}(q^n, \tilde{q})$, which is a second-order TPBVP, but can be heavy to solve. Instead, assuming that the intrinsic curvature of $\mathcal{S}$ in a neighborhood of q^n containing $\tilde{q}$ is not too large, making the metric slowly varying, we can approximate $\gamma_{\mathcal{S}}(q^n, \tilde{q})$ by the straight-line segment (the dotted line of the figure) joining q^n and $\tilde{q}$. We then project the vector $\overrightarrow{q^n \tilde{q}}$ to $T_{q^n} V$, this projection being denoted by $\mathbf{w}^n$. Now that we have an estimate of the error at q^n on the form of the vector $\mathbf{w}^n$, we need to carry this information back to the initial location q. This is done by parallel

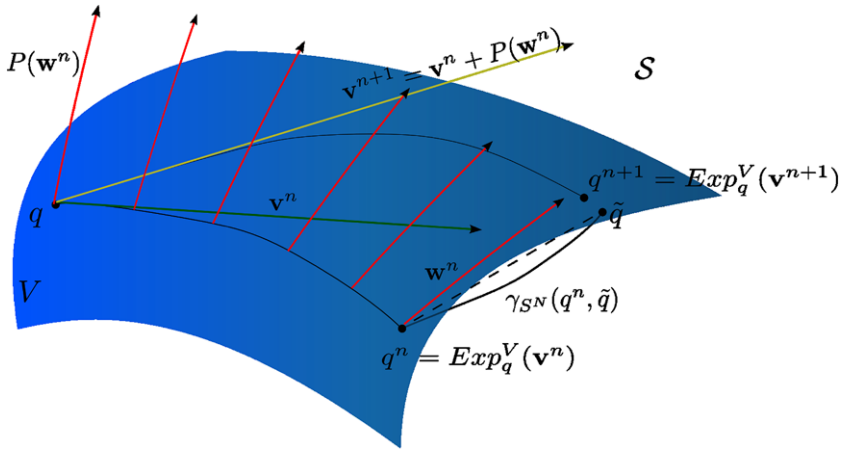


Fig. 1 A step in Log map/geodesic computation by a shooting method. q is the start point and $\tilde{q}$ the end point ones tries to reach. See the text for further explanations

Table 1 Algorithm to compute the Riemannian Log map on the subspace V

Compute $Log_{x_1}(x_2)$

1. Set $v = \Pi^G_{T_{x_1}V}(x_2 - x_1)$
2. Compute $\bar{x} = Exp_{x_1}(v)$
3. If $|x_2 - \bar{x}| < \epsilon$, return v, else continue.
4. Compute $\bar{v} = \Pi^G_{T_{\bar{x}}V}(x_2 - \bar{x})$
5. Compute v' the Parallel Transport of $\bar{v}$ to $T_{x_1}V$, using (21).
6. Update $v = v + v'$ and go to Step 2.

transporting it back along the geodesic $\gamma_V(q, q^n)$ in V joining q and q^n. The parallel transport $P(\mathbf{w}^n)$ is computed via the covariant derivative $\frac{D^V}{dt}$ of $\gamma_V(q, q^n)$ in V. Once done, we update the initial velocity $\mathbf{v}^{n+1} = \mathbf{v}^n + P(\mathbf{w}^n)$.

Such a procedure is generally not guaranteed to converge, it will depend on the intrinsic curvatures of $\mathcal{S}$ and V in a neighborhood of q and $\tilde{q}$ as well as the extrinsic curvature of V in $\mathcal{S}$ (generalizing the second fundamental form). When manifolds are sufficiently "flat" it generally work well however. The algorithm is given in Table 1. A fundamental step is computing the parallel transport of a vector $\mathbf{w}(0) \in T_{x(0)}V$ along a geodesic curve $x(t) \in V$. We now derive the needed equations. The parallel transport is characterized by

$$\mathbf{w}(t) \in T_{x(t)}V \quad \Rightarrow \quad DF(x(t), \mathbf{w}(t)) = 0, \tag{15}$$

$$\frac{D^V}{dt}\mathbf{w}(t) = 0 \quad \Rightarrow \quad \frac{D^{\mathcal{S}}}{dt}\mathbf{w}(t) \in N_{x(t)}V \tag{16}$$

where $\frac{D^V}{dt}$, $\frac{D^{\mathcal{S}}}{dt}$ are the covariant derivatives with respect to the space V and the embedding manifold $\mathcal{S}$ respectively. Differentiating (15) and re-arranging terms we get

$$\frac{d\mathbf{w}}{dt} = -(JF(x))^{\dagger}((D^2F(x)\dot{x})\mathbf{w}) \tag{17}$$

where $JF(x)^{\dagger}$ is the G-pseudo-inverse of $JF(x)$ and $D^2F(x)$ is the second differential of F at x. $JF(x)^{\dagger}$ is computed using the method described in previous section, which gives

$$JF(x)^{\dagger} = G^{-1}JF(x)^T(JF(x)G^{-1}JF(x)^T)^{-1},$$

(16) gives

$$\frac{D^{\mathcal{S}}}{dt}\mathbf{w}(t) = \sum_k \left(\frac{dw_k}{dt} + \sum_{ij}\Gamma^k_{ij}w_i\dot{x}_j\right)\frac{\partial}{\partial x_k} \in N_xV \tag{18}$$

where Γ are the Christoffel symbols of the embedding space $\mathcal{S}$. Using the classical relation

$$Null(A)^{\perp} = Range(A^*)$$

for a linear operator A between Euclidean spaces, we get that $N_xV = Range(JF(x)^*)$ and thus there exists $\lambda(t)$ such that

$$\frac{D^{\mathcal{S}}}{dt}\mathbf{w}(t) = JF(x)^*\lambda(t) \tag{19}$$

where the adjoint $JF(x)^*$ is given by

$$JF(x)^* = G^{-1}JF(x)^T.$$

Let $f(\mathbf{w}(t), \Gamma, \dot{x}(t)) = [\sum_{ij}\Gamma^k_{ij}w_i\dot{x}_j]_{k=1,\ldots,N}$. Rewriting (18) and (19), we get

$$\frac{d\mathbf{w}}{dt} + f(\mathbf{w}(t), \Gamma, \dot{x}(t)) = JF(x)^*\lambda(t). \tag{20}$$

Put (17) in (20) to compute $\lambda(t)$:

$$\lambda(t) = (JF(x)^*)^{\dagger}(f(\mathbf{w}(t), \Gamma, \dot{x}) - JF(x)^{\dagger}((D^2F(x)\dot{x})\mathbf{w}))$$

One can then compute $\frac{d\mathbf{w}}{dt}$ as

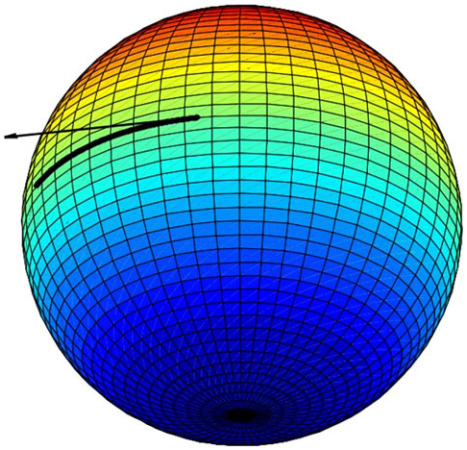


Fig. 2 Exponential map at $x = (0, 1, 0)$ for $v = (1, 0, 0)$ on V_1

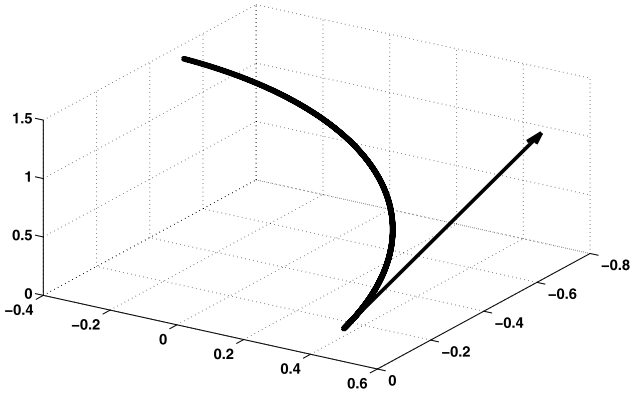


Fig. 3 Exponential map at $x = (0, 0.5, 0.282444)$ for $v = (-0.7039, 0.03, 0.8249)$ on V_2

$$\begin{aligned}\frac{d\mathbf{w}}{dt} &= -f(\mathbf{w}(t), \Gamma, \dot{x}) \\ &\quad + (JF(x)^*)(JF(x)^*)^\dagger(f(\mathbf{w}(t), \Gamma, \dot{x}) \\ &\quad - JF(x)^\dagger((D^2F(x)\dot{x})\mathbf{w}). \end{aligned} \tag{21}$$

We use a finite difference scheme to implement the above equation.

2.4 Stereographic Projection

We now implement the Exponential map on a toy example and verify the results before moving on to the more complicated curve evolution equations with the L_2 metric. For this example, we have $\mathcal{S} = \mathbb{R}^3$, $V = F^{-1}(0)$, $F : \mathbb{R}^3 \to \mathbb{R}$. The metric G is the one induced by stereographic projection $P : \mathbb{S}^3 \backslash (0, 0, 0, 1) \to \mathbb{R}^3$ which is given by

$$P(x_1, x_2, x_3, x_4) = \left(\frac{x_1}{1 - x_4}, \frac{x_2}{1 - x_4}, \frac{x_3}{1 - x_4}\right).$$

Set $\ell := 1 + y_1^2 + y_2^2 + y_3^3$. Then the inverse of P is

$$P^{-1}(y_1, y_2, y_3) = \left(\frac{2y_1}{\ell}, \frac{2y_2}{\ell}, \frac{2y_3}{\ell}, \frac{\ell - 2}{\ell}\right).$$

Then, the corresponding inner product induced on $\mathbb{R}^3$ at a point $\mathbf{y} = (y_1, y_2, y_3)$ is [30]

$$G_{\mathbf{y}}(v_1, v_2) = \frac{4}{\ell^2}\langle v_1, v_2\rangle_{\mathbb{R}^3} \tag{22}$$

(note that this is a classical example of a conformal equivalence). Let $F_1(\mathbf{y}) = y_1^2 + y_2^2 + y_3^2 - 1$ and $V_1 = F_1^{-1}(0)$. This is the equatorial great circle of $\mathbb{S}^3$. The gradient and Hessian of F and DG_x are straightforward to compute. Figure 2 shows the result of our exponential map algorithm. With this particular choice of F, the situation might be a bit too "toyish". Figure 3 shows the exponential map on $V_2 = F_2^{-1}(0)$ for $F_2(\mathbf{y}) = 20y_1 + 2y_2 + 20y_3 - 5y_1^2 - 5y_2^2 - 5x_3^2 - 5$, which is the intersection of a tilted plane passing through the origin and $\mathbb{S}^3$.

We can verify whether the obtained curves are geodesics or not by checking the tangential component of the acceleration vector. The acceleration vector (in the embedded space) is given by the covariant derivative (for the embedding space) of the velocity vector field (along itself). This is required since the metric on the embedding space is not the usual Euclidean one, which gives rise to non-zero Christoffel symbols (Christoffel symbols and covariant derivative definitions are recalled in Appendix). For a curve γ of V_i, $i = 1, \ldots, 2$, to be geodesic, we must have

$$\ddot{\gamma}(t) = \nabla_{\dot{\gamma}(t)}\dot{\gamma}(t) \in \mathcal{N}_{\gamma(t)}V_i$$

where $\mathcal{N}_{\gamma(t)}V_i$ is the orthogonal complement of $T_{\gamma(t)}V_i$ in $T_{\gamma(t)}\mathcal{S}$. The tangential component of the acceleration in V_i is computed by projecting the covariant derivative $\frac{D\dot{\gamma}}{dt}$ of the velocity vector $\dot{\gamma}$ to the tangent space. Figures 4 and 5 show the norm of different accelerations and projected accelerations for the two geodesics computed on V_1 and V_2 respectively. We also found that the error between the analytically computed geodesic and the geodesic computed with our algorithm reduces as we reduce the time-step, as shown in Fig. 6.

We have also verified our algorithm for computing Log map on both V_1 and V_2. A key ingredient being parallel transport, Fig. 7, illustrates the parallel transport of a vector along a geodesic in V_2. It should, up to solver's precision have zero normal component and covariant derivatives. We show the norms of the normal components and covariant derivatives in Fig. 8. Although errors accumulate, norm of the normal component remains below 6×10^{-4} after 1000 iterations and the norm of the covariant derivative is even smaller.

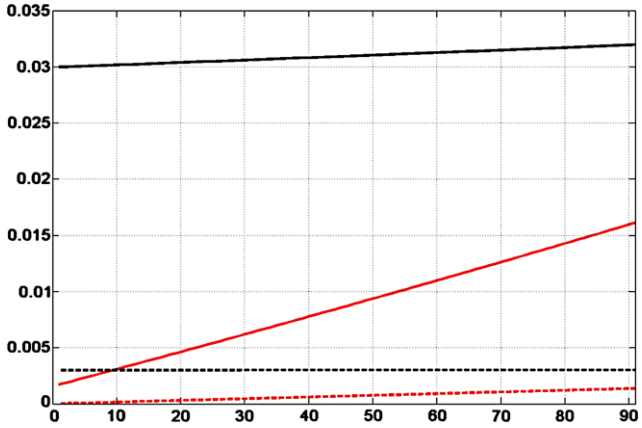


Fig. 4 (Color online) *The black* and *red curves* are the norms of tangential and normal acceleration vectors of the geodesic on V_1, respectively. *The solid curves* are computed with a time step of 0.01, while *the dashed curves* are with a time step of 0.001. Due to numerical errors, the tangential component is larger than the normal component of acceleration. Still the magnitudes are the order of 10^{-2}. One can also observe that the tangential and normal component reduce as we reduce the time step, thereby converging to zero, the desired theoretical value

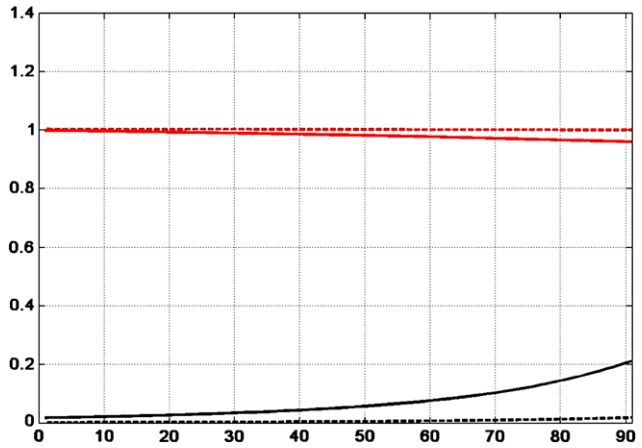


Fig. 5 (Color online) *The black* and *red curves* are the norm of tangential and normal acceleration vector of the geodesic on V_2, respectively. *The solid curves* are computed with a time step of 0.01, while *the dashed curves* are with a time step of 0.001. One can observe that the tangential component reduces as we reduce the time step of the algorithm

3 Subspace of Equidistant Neighboring Node Point Spline Curves

We recall first a few elementary facts on cardinal B-splines, and refer to [4, 31–33] for details.

Set $\beta^0(x) = \chi_{[-\frac{1}{2},\frac{1}{2})}$ the 0-th order basis spline, the n-th order B-splines is defined inductively as

$$\beta^n(x) = \underbrace{\beta^0(x) * \beta^0(x) * \cdots * \beta^0(x)}_{n+1 \text{ times}}$$

where $*$ denote the convolution product.

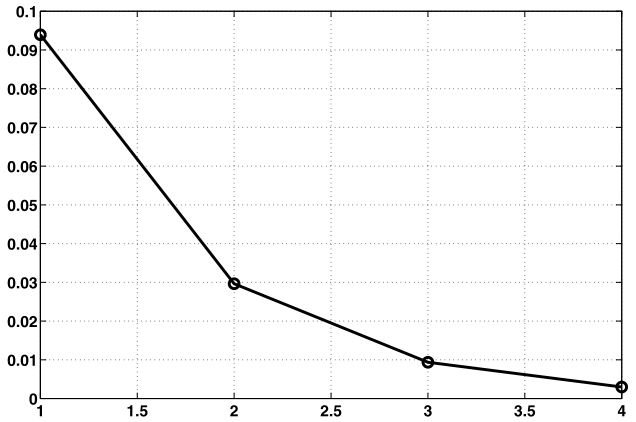


Fig. 6 Plot of sup norm of the difference between the numerically computed geodesic shown in Fig. 2 and analytically computed geodesic on V_1 against the time-step used in our algorithm

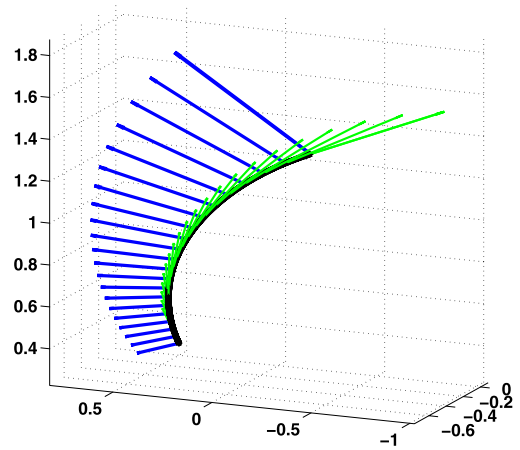


Fig. 7 (Color online) Parallel transport of a vector $w \in T_{x(0)}V$ along previously computed geodesic on V_2. The transported vectors are shown in *blue* and tangent to the geodesic in *green*

Fixing such an order and an integer $N > 0$, we consider the space of curves

$$c(s) = \sum_{i \in \mathbb{Z}} \beta^n(s-i)c_i, \quad c_i = (c_i^x, c_i^y) \in \mathbb{R}^2$$

with the periodicity condition

$$c_{i+N} = c_i.$$

The periodicity in the sequence of control points insures that the curve c is closed and everywhere of class C^{n-1}. The space of such curves is denoted as S^{2N} and is obviously of dimension $2N$, with the following basis

$$e_{1i} = \begin{pmatrix} \beta^n(s-i) \\ 0 \end{pmatrix}, \qquad e_{2i} = \begin{pmatrix} 0 \\ \beta^n(s-i) \end{pmatrix},$$

$$i = 1, \ldots, N.$$

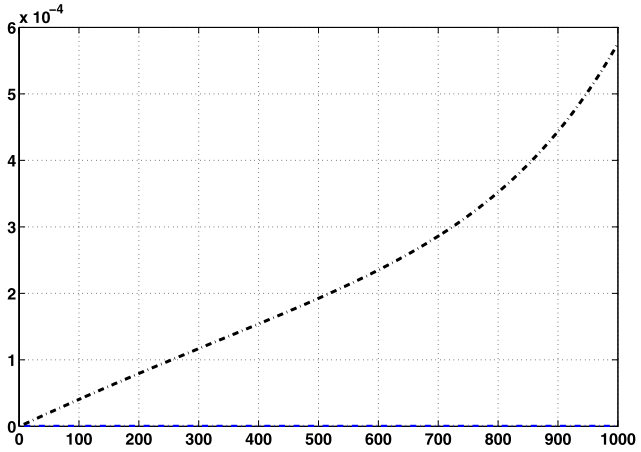


Fig. 8 (Color online) The norm of the normal component of the parallel transported vector shown in Fig. 7 is *the black curve* ($\simeq 10^{-4}$) and covariant derivative in V_2 of the parallel transported vector is shown *in blue* ($\simeq 0$)

This is not an orthogonal basis, as, though

$$\langle e_{1i}, e_{2j}\rangle = 0,$$

$$\langle e_{ki}, e_{kj}\rangle = \int_0^N \beta^n(s-i)\beta^n(s-j)\,dx = \delta_{n,|i-j|}$$

depends on n and the distance between i and j.

Thus the basis vectors $(e_{ki})_{i=1,\ldots,N}^{k=1,2}$ are not orthogonal with respect to the L^2 inner product.

Given such a curve c, at each knot $j \in \{1,\ldots,N\}$, corresponds a "node point" $P_j = (P_j^x, P_j^y) = c(j)$(not c_j, which denotes a control point), and this correspondence is *unique and linear*, given by an $N \times N$ matrix M_N^n

$$\begin{pmatrix} c_1^x & c_1^y \\ \vdots & \vdots \\ c_N^x & c_N^y \end{pmatrix} = M_N^n \begin{pmatrix} P_1^x & P_1^y \\ \vdots & \vdots \\ P_N^x & P_N^y \end{pmatrix}$$

see [4] for details.

We generally start with initial node points and build spline curves passing through them, curves that then will evolve. In most of the curve evolution algorithms we will discuss, we need to be able to compute first and second order derivatives of our spline curves, thus β^n should be of order at least 3. In the sequel we assume the order n fixed as well as the number N of control points/node points. We now define the constraints, as follows. Set

$$d_{i,j} = \|P_j - P_i\|_2^2$$

the squared Euclidean distance between two points P_i, P_j in $\mathbb{R}^2$. Then the subspace can be written as simple quadratic constraints $F : S^{2N} \to \mathbb{R}^{N-1}$ given as

$$F_i(P_1,\ldots,P_N) = d_{i+2,i+1} - d_{i+1,i}, \quad i = 1,\ldots,N-2,$$

$$F_{N-1}(P_1,\ldots,P_N) = d_{1,N} - d_{N,N-1}. \tag{23}$$

The configuration space is the subspace of S^{2N} given by $\tilde{A}_N = F^{-1}(0)$. The tangent space of $\tilde{A}_N$ at a configuration x is given by

$$T_x\tilde{A}_N = \ker(JF(x)) \tag{24}$$

the kernel (or null space) of the Jacobian of F at point $x \in \tilde{A}_N \subset S^{2N}$. We may call $\tilde{A}_N$ a N-links bicycle chain manifold. We have explored it in a previous paper [34]. As mentioned in Sect. 2, the hierarchy of the subspaces in this case is more complicated. Not all curves in the space of closed spline curves with N control points (S^{2N}) are immersions. Moreover the group of circular permutations on node points has no 'geometric' effect on the spline curve. This will result in a non-zero distance between a curve and a curve obtained by circular permutation of the set of node points of the original curve. We overcome this problem simply by choosing a corresponding starting point on the curves. We let $Imm^N = S^{2N} \cap B_i$. Similarly not all curves in $\tilde{A}_N$ are immersions. Also the Jacobian of F may not be full rank everywhere on S^{2N}. We let $A'_N \subset \tilde{A}_N$ be the subset where the Jacobian is full rank and let $A_N = A'_N \cap Imm^N$ be the space of equidistant neighboring node point closed spline curves that are immersions and where the Jacobian is full rank. For now, given a curve in A_N, we hope that we stay in A_N along an evolution, without putting up additional constraints to ensure that a curve remains an immersions and that the rank of the Jacobian does not reduce (we use $T_xA_N = ker(JF(x))$), accepting the fact that time and again we may end up in the space $\tilde{A}_N \setminus A'_N$.

We now need to construct an inner product on A_N. Instead of restricting the scalar product of $\mathbb{R}^{2N}$ to T_xA_N, we induce the L_2 metric of B_i and S^N on T_xA_N.[4] We now derive the expression for the L_2 inner product on the spline subspace. Given two deformations $p(s) = \sum_i \beta^n(s-i)p_i$ and $q(s) = \sum_i \beta^n(s-i)q_i$ of the spline curve $c(s)$, the inner product of the deformations is given as

$$\begin{aligned}\langle p(s), q(s)\rangle_c &= \int_0^N \left(\sum_i \beta^n(s-i)p_i\right)\cdot n(s) \\ &\quad \times \left(\sum_j \beta^n(s-j)q_j\right)\cdot n(s)|c'(s)|\,ds \\ &= P^T G(c) Q\end{aligned}$$

where $a \cdot b$ is the usual dot product between two vectors a and b, $n(s)$ is the inner unit normal to the curve $c(s)$

[4] Not all N node points spline curves are immersions. But we assume that the evolution velocity keeps the curve in B_i.

and $P = (p_x^0, p_y^0, \ldots, p_x^{N-1}, p_y^{N-1})$ and $Q = (q_x^0, q_y^0, \ldots, q_x^{N-1}, q_y^{N-1})$ are the control point vectors. The Gram matrix $G(c)$ is a $2N \times 2N$ matrix which can be written as an $N \times N$ matrix of smaller 2×2 matrices given by

$$G_{ij}(c) = \int_0^N \beta^n(s-i)\beta^n(s-j)n(s)n(s)^T |c'(s)|\, ds.$$

One can clearly see the smoothing effect of the splines in the Gram matrix. This limits the high frequency variation of a deformation.

The derivative of the Gram matrix with respect to the node points P_k of the curve $c(s)$ at which it is computed is a $2N \times 2N \times 2N$ matrix whose members are $2N \times 2N$ matrices of the form

$$\frac{dG_{ij}}{dP_k}(c) = \int_0^N \beta^n(s-i)\beta^n(s-j) \times \frac{d}{dP_k}(n(s)n(s)^T |c'(s)|)\, ds. \tag{25}$$

The inner unit normal of a curve $c(s) = (x(s), y(s))$ is given as $n(s) = \frac{(-y'(s), x'(s))^T}{\sqrt{x'(s)^2 + y'(s)^2}}$. Given the direct and indirect spline transforms [32, 33], one can compute the derivatives in (25) with little work. The gradient and Hessian of the equidistant constraint are easy to compute and the pseudoinverse of the $J_G F^T$ matrix can be computed as shown in (14).

To restrict a given curve evolution to the subspace A_N, we simply iterate the three projection steps described in Sect. 2. The projection to S^{2N} is given by (using (6))

$$v_S(t) = \sum_j \left[G_1^{-1} \left(\int_0^N e_i(s) v(s,t) \times n(s,t)|c'(s)|\, ds \mid_{i=1\ldots N} \right) \right]_j e_j \tag{26}$$

where e_i are the spline basis vectors of S^{2N}. One can observe that computation of the motion of a control point due to the velocity $v(s,t)n(s,t)$ is non-local, i.e., depending on the basis vectors (splines in this case), the control point motion computation is a weighted average like process over a certain neighborhood. This comes from the fact that finite dimensional curve representation has to be non-local in nature. The spline basis vectors filter out high frequency variations of the velocity field on the curve, which in some sense gives similar effect to using the Sobolev inner product on the space of curves (as in [9]). The projection to $T_{c(t)}A_N$ is given by (6) and we can further project the obtained tangent velocity vector to the N-links bicycle chain manifold using the Exponential map given by (11), (12), (13) and (14).

4 Experiments

We verify our algorithm to restrict curve evolution and to compute Log maps with the following experiments:

4.1 Curvature Flow

Curvature flow is given as

$$\frac{\partial c}{\partial t}(p,t) = \kappa(p,t)n(p,t) \tag{27}$$

where $\kappa(-,t)$ is the curvature of the curve $c(-,t)$. This flow has interesting properties like

- A simple curve, under the curvature flow, evolves into a convex curve and vanishes at a round point [35].
- Under the curvature flow, the curve does not self intersect [35].
- The curvature flow is a curve shortening flow, i.e., it is a gradient descent on the length of the curve.
- Two curves under curvature flow, one completely contained in the other, will not intersect during their evolution.
- The curvature flow simplifies curves, i.e., the variation of curvature of a curve reduces with time [2].

We project this flow to the N-links bicycle chain manifold. We would like this projected flow to preserve as many of the properties of the original flow as possible. Empirically, one can see that *at least* some of these properties are preserved, see Fig. 9. Specifically, the curve evolves into a curve that is as close as possible to a circle in the N links bicycle chain manifold. Also observe that there is a tangential motion to the curve in order to satisfy the implicit constraints.

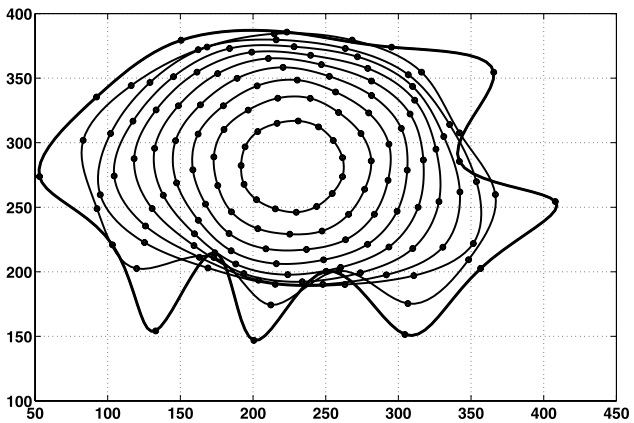


Fig. 9 Curvature flow of a non-convex curve. The points marked by 'o' are node points of the spline curve. Observe that there is a tangential motion to the curve in order to satisfy the equidistant constraint

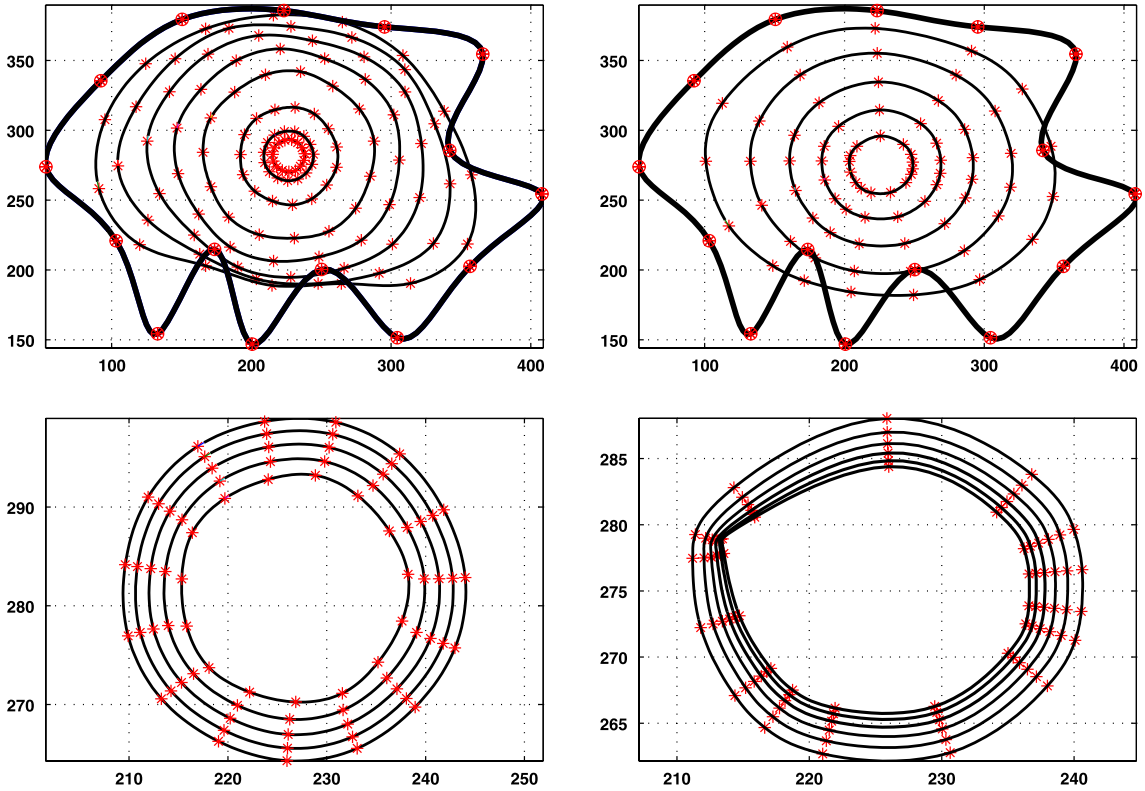


Fig. 10 (*Top-left*) Curvature flow restricted to N-links bicycle chain manifold. (*Top-right*) Curvature flow restricted to linear spline subspace. (*Bottom-left*) some of the final iterations of curvature flow in N-links bicycle chain manifold. (*Bottom-right*) some of the final iterations of curvature flow in spline subspace. Notice the numerical errors arising because of node points coming too close to each other

4.2 Comparison with Unconstrained Flow

The node points of a spline curve may come too close to each other during the evolution. But when the evolution is restricted to the N links bicycle chain manifold, the node points remain equidistant from each other by construction. Hence the evolution is numerically more stable. In Fig. 10, we compare the curvature flow restricted to the N links bicycle chain manifold with the curvature flow restricted to the linear spline subspace. We see that the node points can get very close to each other leading to numerical instability.

4.3 Geodesic Active Contours

Geodesic active contours are used to segment objects from an image. The user defines an initial curve around the object of interest, which then evolves and adheres to the object boundary. The curve evolution equation is given as

$$\frac{\partial c}{\partial t}(p,t) = (\kappa g - \langle \nabla g, n\rangle + \alpha g)n \tag{28}$$

where $g : \mathbb{R} \to \mathbb{R}$ is the edge detection function

$$g(|\nabla I(c(p))|) = \frac{1}{1+|\nabla I(c(p))|^2}$$

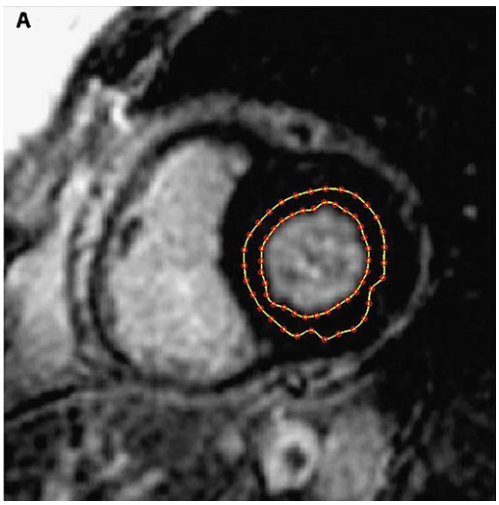


Fig. 11 (Color online) Geodesic active contours on a heart image. The points marked with *red* 'o' are the node points

and the αg term helps in detecting non-convex objects [36]. We show a segmentation result on a heart MRIimage in Fig. 11 in which the contours are members of the N links bicycle chain manifold.

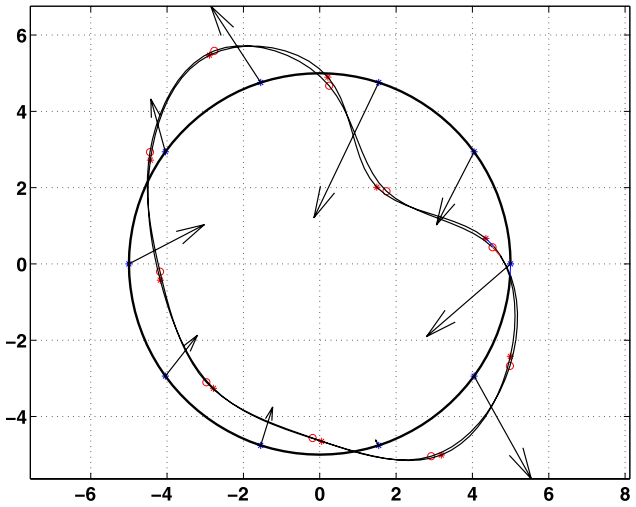


Fig. 12 Log of the deformed kidney shaped curve (with node points marked by '*') with a circular curve as the base point is shown by deformation vectors on the node points of the circle. The other kidney shaped curve (with node points markes with 'o') is the Exponential map of the computed Log map at the base point (*circle*)

4.4 Log Map

The Log map between two curves defines a vector field on the first curve. We use shooting method to compute the Log map. The vector field, two initial curves, and the Exponential map of the computed Log map are shown in Fig. 12. Since there is a threshold parameter in the shooting method, the Exponential map of the computed Log map does not exactly match the target curve. One can obtain more accurate results by lowering the threshold parameter.

5 Conclusion and Future Work

In this paper we give a new algorithm to restrict curve evolution to implicitly defined subspace induced with a non-Euclidean metric. We give a new algorithm to compute the Exponential map on implicitly defined manifolds equipped with a non-Euclidean metric. The N-links bicycle chain manifold is a useful space to work in, specially in medical imaging, where landmarks and pseudo-landmarks are often used. Moreover this algorithm can be used as an optimization method to optimize over implicitly defined manifolds with a given inner product.

One of the drawbacks of this method is that as the number of points on the curve increases, computing the derivative of the Gram matrix $\frac{dG}{dX}$ becomes very expensive. Currently we use an iterative shooting scheme to compute the Riemannian Log map which may not always converge. We need to look for more general schemes for the Log map.

Another direction of future work is to understand the behaviour of geodesics when the number N of node points tends to $+\infty$. Will they exhibit severe high frequency oscillations? This was not observed in our experiments, where $N \approx 30$, and this raises the related question of how many node points are needed for a given problem.

Appendix: Definitions from Differential Geometry

We give definitions of some concepts from differential geometry that we use in the paper (mainly from [21]) for the convenience of the reader.

1. Differentiable Manifolds:
   A differentiable manifold of dimension n is a set M and a family of injective mappings $\mathcal{T} = \{x_i : U_i \subset \mathbb{R}^n \to M\}$ of open sets U_i of $\mathbb{R}^n$ into M such that
   - $\bigcup_i x_i(U_i) = M$, i.e. the open sets cover M.
   - For any pair i, j with $x_i(U_i) \cap x_j(U_j) = W \neq \phi$, the mapping $x_j^{-1} \circ x_i$ is differentiable.
   - The family $\mathcal{T}$ is maximal, which means that if (y, V), $y : V \subset \mathbb{R}^n \to M$ is such that: for each element of $\mathcal{T}$, (x_i, U_i) with $x_i(U_i) \cap y(V) \neq 0$ implies that $y^{-1} \circ x_i$ is a diffeomorphism, then in fact $(y, V) \in \mathcal{T}$.
2. Directional derivative of a function along a vector field:
   A vector field X on M is a map that associates to each $p \in M$ an element $X(p) \in T_pM$, where T_pM is the tangent space of M at p. Let $f : M \to \mathbb{R}$ be a differentiable function of M and X a vector field on M. The directional derivative $X(f)$ is the function $M \to \mathbb{R}$,
   $$X(f)(p) = df_p(X(p))$$
   the differential of f at p evaluated at vector $X(p)$.
3. Immersion and Embedding:
   A differentiable mapping $\psi : M \to N$ between two manifolds is an immersion if its differential map $d\psi_p : T_pM \to T_{\psi(p)}N$ is injective for all $p \in M$. If ψ is also a homeomorphism from M onto $\psi(M) \subset N$ where $\psi(M)$ has the subspace topology of N, then ψ is called an embedding and $\psi(M)$ is an (embedded) submanifold of N.
4. Riemannian Metric:
   A Riemannian metric on a manifold M is a correspondence which associated to each point $p \in M$ an inner product $\langle -, - \rangle_p$ on the tangent space T_pM, which varies smoothly. In terms of local coordinates, the metric at each point x is given by a matrix, $g_{ij} = \langle X_i, X_j \rangle_x$, where X_i, X_j are tangent vectors to M at x, and it varies smoothly with x. A *Geodesic curve* is a local minimizer of arc-length computed with a Riemannian metric.
5. Affine connection:
   Let $\mathcal{X}(M)$ be the set of all smooth vector fields on M.

An affine connection ∇ on a differentiable manifold M is a mapping

$$\nabla : \mathscr{X}(M) \times \mathscr{X}(M) \to \mathscr{X}(M)$$

which is denoted by $\nabla(X, Y) \to \nabla_X Y$ and which satisfies the following properties:

- $\nabla_{fX+gY} Z = f\nabla_X Z + g\nabla_Y Z$
- $\nabla_X(Y + Z) = \nabla_X Y + \nabla_X Z$
- $\nabla_X(fY) = f\nabla_X Y + X(f)Y$

in which $X, Y, Z \in \mathscr{X}(M)$ and f, g are $\mathcal{C}^\infty(M)$. This gives us a notion of directional derivative of a vector field defined on the manifold.

6. Covariant derivative:
Let M be a differentiable manifold with affine connection ∇. There exists a unique correspondence which associates to a vector field V along the differentiable curve $c : I \to M$ another vector field $\frac{DV}{dt}$ along c, called the covariant derivative of V along c, such that
   - $\frac{D}{dt}(V + W) = \frac{DV}{dt} + \frac{DW}{dt}$, where W is a vector field along c.
   - $\frac{D}{dt}(fV) = \frac{df}{dt}V + f\frac{DV}{dt}$, where f is a differentiable function on I.
   - If V is induced by a vector field Y, a member of the tangent bundle of M, i.e. $V(t) = Y(c(t))$, then $\frac{DV}{dt} = \nabla_{\frac{dc}{dt}} Y$.

   In a parametrized manifold, where the curve $c(t)$ is represented as $(x^1(t), \ldots, x^n(t))$, the covariant derivative becomes

$$\frac{Dv}{dt} = \sum_k \left\{ \frac{dv^k}{dt} + \sum_{i,j} \Gamma^k_{ij} v^j \frac{dx^i}{dt} \right\} \frac{\partial}{\partial x_k} \tag{29}$$

   where the Γ^k_{ij} are the *coefficients of the connection* also known as the *Christoffel symbols* Γ. For the Levi-Civita connection associated with the metric g of a Riemannian manifold, the corresponding Christoffel symbols are given by

$$\Gamma^k_{ij} = \frac{1}{2} \sum_m \left\{ \frac{\partial}{\partial x_i} g_{jm} + \frac{\partial}{\partial x_j} g_{mi} - \frac{\partial}{\partial x_m} g_{ij} \right\} g^{mk} \tag{30}$$

   g_{ij} is the ijth element of the metric, and g^{ij} is the ijth element of its inverse. A curve is geodesic if the covariant derivative of its tangent vector field is zero everywhere on it, which means that a geodesic curve has zero tangential acceleration. Such a curve c satisfies the second order system of ODEs, which, with the above parametrization becomes

$$\frac{d^2x^k}{dt^2} + \sum_{ij} \Gamma^k_{ij} \frac{dx^i}{dt} \frac{dx^j}{dt} = 0, \quad k = 1, \ldots, n. \tag{31}$$

7. Exponential map:
The exponential map is a map $Exp : TM \to M$, that maps $v \in T_qM$ for $q \in M$, to a point on M obtained by going out the length equal to $|v|$, starting from q, along a geodesic which passes through q with velocity equal to $\frac{v}{|v|}$. Given $q \in M$ and $v \in T_qM$, and a parametrization $(x_1, \ldots, x_n)$ around q, $Exp_q(v)$ can be defined as the solution at time 1 of the above system of ODEs (31) with initial conditions $(x^i(0)) = q$ and $(\frac{dx^i}{dt}(0)) = v$, $i = 1, \ldots, n$. The geodesic starting at q with initial velocity t can thus be parametrized as

   $$t \mapsto Exp_q(tv).$$

8. Log map: For $\tilde{q}$ in a sufficiently small neighborhood of q, the length minimizing curve joining q and $\tilde{q}$ is unique as well. Given q and $\tilde{q}$, the direction in which to travel geodesically from q in order to reach $\tilde{q}$ is given by the result of the logarithm map $Log_q(\tilde{q})$. We get the corresponding geodesics as the curve $t \mapsto Exp_q(tLog_q\tilde{q})$. In other words, *Log* is the inverse of *Exp* in the neighborhood.

References

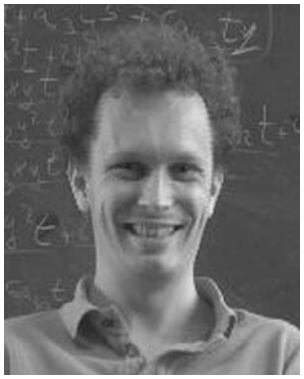
1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. Int. J. Comput. Vis. 321–331 (1987)
2. Sethian, J.A.: Level Set Methods, 1st edn. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, Cambridge (1996)
3. Kimmel, R.: Numerical Geometry of Images, Theory, Algorithms and Applications, 1st edn. Springer, New York (2004). Department of Computer Science, Technion-Israel Institute of Technology, Haifa 32000, Israel
4. Blake, A., Isard, M.: Active Contours. Springer, Berlin (1998)
5. Epstein, C.L., Gage, M.: The curve shortening flow. In: Wave Motion: Theory, Modeling and Computation. Springer, New York (1987)
6. Yezzi, A.J., Mennucci, A.: Metrics in the space of curves. Technical Report (2005). arXiv:math.DG/0412454
7. Charpiat, G., Keriven, R., Pons, J., Faugeras, O.: Designing spatially coherent minimizing flows for variational problems based on active contours. In: Tenthf ICCV, vol. 2, Beijing, China. IEEE, New York, pp. 1403–1408 (2005)
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. Int. J. Comput. Vis. **22**, 61–79 (1997)
9. Sundaramoorthi, G., Yezzi, A., Mennucci, A.C.: Sobolev active contours. Int. J. Comput. Vis. **73**, 345–366 (2007)
10. Charpiat, G., Maurel, P., Pons, J.P., Keriven, R., Faugeras, O.: Generalized gradients: priors on minimization flows. Int. J. Comput. Vis. **73**, 325–344 (2007)
11. Michor, P.W., Mumford, D.: Riemannian geometries on spaces of plane curves. J. Eur. Math. Soc. **8**, 1–48 (2006)
12. Sundaramoorthi, G., Mennucci, A., Soatto, S., Yezzi, A.: Tracking deforming objects by filtering and prediction in the space of curves. CDC (2009)
13. Glaunés, J., Qui, A., Miller, M.I.: Large deformation diffeomorphic metric curve mapping. Int. J. Comput. Vis. **80**, 317–336 (2008)

14. Miller, M.I., Younes, L.: Group actions, homeomorphisms, and matching: a Cambridge general framework. Int. J. Comput. Vis. **41**, 61–84 (2001)
15. Younes, L.: Computable elastic distances between shapes. SIAM J. Appl. Math. **58**, 565–586 (1998)
16. Micheli, M.: The differential geometry of landmark shape manifolds: metrics, geodesics and curvature. PhD thesis, Brown University, Providence, Rhodes Island (2008)
17. Unal, G., Yezzi, A., Krim, H.: Information-theoretic active polygons for unsupervised texture segmentation. Int. J. Comput. Vis. **62**, 199–220 (2005)
18. Srivastava, A., Mio, W., Klassen, E., Liu, X.: Geometric analysis of constrained curves for image understanding. In: Proc. 2nd IEEE International Workshop on Variational, Geometric and Level-Set Methods in Computer Vision (VLSM) (2003)
19. Tatu, A., Lauze, F., Nielsen, M., Olsen, O.F.: Curve evolution in subspaces. In: Scale Space and Variational Methods in Computer Vision (2007)
20. Hairer, E.: Geometric integration of ordinary differential equations on manifolds. BIT Numer. Math. **41**, 996–1007 (2001)
21. do Carmo, M.P.: Riemannian Geometry. Mathematics: Theory and Applications. Birkhäuser, Basel (1992)
22. Luenberger, D.G.: Optimization by Vector Space Methods. Wiley, New York (1969)
23. Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V., Mishchenko, E.F.: The Mathematical Theory of Optimal Processes. Interscience, New York (1962)
24. Kamien, M.I., Schwartz, N.L.: Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management. Elsevier, Amsterdam (1991)
25. Dedieu, J.P., Nowicki, D.: Symplectic methods for the approximation of the exponential map and the newton iteration on Riemannian submanifolds. J. Complex. **21**, 487–501 (2005)
26. Gallot, S., Hulin, D., Lafontaine, J.: Riemannian Geometry, 3rd edn. Springer, Berlin (2004)
27. Kamaraj, K., Sivakumar, K.C.: Moore-penrose inverse in an indefinite inner product space. J. Appl. Math. Comput. **19**, 297–310 (2005)
28. Keller, H.B.: Numerical Methods for Two-Point Boundary-Value Problems. Blaisdell, Waltham (1968)
29. Keller, H.B.: Numerical Solution of Two Point Boundary Problems. CBMF-NSF Regional Conference Series in Applied Mathematics, vol. 24. SIAM, Philadelphia (1976)
30. Lee, J.M.: Riemannian Manifolds: An Introduction to Curvature. Springer, Berlin (1997)
31. Unser, M.: Splines: a perfect fit for signal and image processing. IEEE Signal Process. Mag. (1999)
32. Unser, M., Aldroubi, A., Eden, M.: B-spline signal processing: Part 1—theory. IEEE Trans. Signal Process. **41**, 821–832 (1993)
33. Unser, M., Aldroubi, A., Eden, M.: B-spline signal processing: Part 2—efficient design and applications. IEEE Trans. Signal Process. **41**, 834–848 (1993)
34. Sommer, S., Tatu, A., Chen, C., de Bruijne, M., Loog, M., Jorgensen, D., Nielsen, M., Lauze, F.: Bicycle chain shape models. In: CVPR Workshop on Mathematical Methods in Biomedical Image Analysis (2009)
35. Grayson, M.A.: The heat equation shrinks embedded plane curves to round points. J. Differ. Geom. **26** (1987)
36. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing. Springer, Berlin (2006)

Aditya Tatu did his Bachelor of Engineering (Electronics & Communication) from Sardar Patel University, India in 2003. He received his MTech (Info. & Comm. Technology) from Dhirubhai Ambani Institute of Information & Communication Technology (DA-IICT), Gandhinagar, Gujarat, India in 2005. He received his MSc and PhD from the Department of Computer Science (DIKU), University of Copenhagen, Denmark in 2007 and 2010 (April) respectively, under the supervision of Mads Nielsen and Francois Lauze. As part of his PhD he worked at the LEMS lab, Division of Engg, Brown University, USA for a semester in 2008. He is currently an Assistant Professor at DA-IICT, India. His topics of interest include applications of differential geometry and calculus of variations in image analysis (segmentation, shapes, image features).

François Lauze has studied Mathematics at the University of Nice–Sophia-Antipolis where he graduated with a PhD in Algebraic Geometry in 1994, before moving to Ouagadougou, Burkina Faso, where he taught mathematics and then Denmark where he studied Image Analysis at the IT University of Copenhagen and graduated with yet another PhD in 2004. Since 2008 he has been with the Image Group at the University of Copenhagen where he is currently associate professor. His main topics of interest include Image reconstruction and segmentation, shapes and shape statistics, using variational framework, partial differential equations, differential and Riemannian geometry.

Stefan Sommer received his MSc degree in Mathematics in 2008 from the University of Copenhagen. His master thesis concerned the application of geometric analysis to problems in topology with emphasis on extinction time for manifolds evolving under Ricci flow. He is currently a PhD student supervised by Mads Nielsen and Francois Lauze and member of the Image Group at the University of Copenhagen. His main research interests are the application of geometry to non-linear modeling in computer vision and statistics on Riemannian manifolds.



Mads Nielsen received a MSc in 1992 and a PhD in 1995 both in computer science from DIKU, Department of Computer Science, University of Copenhagen, Denmark. During his PhD studies he spent one year 1993–1994 at the Robotvis <http://www.inria.fr/Equipes/ROBOTVIS-fra.html> lab at INRIA, Sophia-Antipolis, France. In the second half of 1995 he was post-doc at the Image Sciences Institute <http://www.isi.uu.nl/> of Utrecht University, The Netherlands. In 1996 he was joint post-doc at DIKU and 3D-Lab <http://www.lab3d.odont.ku.dk/>, School of Dentistry, University of Copenhagen, where he served as assistant professor 1997–1999. April 1999 he became the first associate professor at the new IT University <http://www.itu.dk/> of Copenhagen, and in 2002 he was promoted to full professor. Since 2007 he has returned heading the Image Group at DIKU and the R&D at Nordic Bioscience Imaging A/S. He was general chair of MICCAI 2006 <http://www.miccai2006.dk>, and is among others member of the editorial board of IJCV and JMIV.