The Role of the Essential Manifold in Data Mining – An Introductory Approach

Maria de Fátima Pina[1,2,3](✉) and Marina Alexandra Andrade[1,4]

[1] ISCTE - IUL, University Institute of Lisbon, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
mfdpina@gmail.com, marina.andrade@iscte-iul.pt

[2] ISLA - Santarém, Superior Institute of Management and Administration of Santarém, Rua Teixeira Guedes, 31, 2000-029 Santarém, Portugal

[3] ISR - UC, Institute of Systems and Robotics, Department of Electrical and Computer Engineering, University of Coimbra, Rua Silvio Lima - Pólo II, 3030-290 Coimbra, Portugal
mfdpina@gmail.com

[4] ISTAR - IUL, Information Sciences and Technologies and Architecture Research Center, University Institute of Lisbon, Av. das Forças Armadas, 1649-026 Lisbon, Portugal

Abstract. Interpolating data and the application of data mining techniques in nonlinear manifolds plays a significant role in different areas of knowledge, ranging from computer vision and robotics, to industrial and medical requests, and these growing number of applications have sparked the research interest of the scientific community to these topics. The Generalized Essential manifold, briefly, Essential manifold, consisting of the product of the Grassmann manifold of all k-dimensional subspaces of $\mathbb{R}^n$ and the Lie group of rotations in $\mathbb{R}^n$, for instance, plays an important role in the problem of recovering the structure and motion from a sequence of images, also known as stereo matching, which is a crucial problem in image processing and computer vision. A well-known recursive procedure to generate interpolating polynomial curves in Euclidean spaces is the classical De Casteljau algorithm, which is a simple and powerful tool widely used in the field of Computer Aided Geometric Design, particularly because it is essentially geometrically based. This algorithm has been generalized to geodesically complete Riemannian manifolds. Thus, having this in mind, in this work we present all the ingredients for a detailed implementation of the generalized De Casteljau algorithm to generate geometric cubic polynomials in the Essential manifold preparing the ground to solve different real interpolation problems in this manifold.

Keywords: Cubic polynomials · Essential manifold · De Casteljau algorithm · Geodesics · Data mining · Interpolating data

O. Gervasi et al. (Eds.): ICCSA 2023 Workshops, LNCS 14112, pp. 173–183, 2023.
https://doi.org/10.1007/978-3-031-37129-5_15

1 Introdution

Interpolation problems involving data on nonlinear manifolds is a topic that appears in a vast number of applications such that face recognition, biomedical image analysis, automobile safety, fingerprint recognition, and in many other problems where large sets of data allow the extraction of important information and identification of patterns.

For illustration, consider the problem of recovering structure and motion from a sequence of images, also known as stereo matching, which is a key problem in computer vision that continues to be one of the most active research areas with remarkable progress in imaging and computing. We refer, for instance, [3,10,15] and references therein for details concerning multiple applications.

The mentioned examples served as a motivation for our study and the main goal of this work is to present an approach for solving interpolation problems on the Generalized Essential manifold highlighting its contribution to data mining.

This manifold is the product of the Grassmann manifold $G_{k,n}$, consisting of all k-dimensional linear subspaces of the real n-dimensional Euclidean space, and the Lie group of rotations $\mathrm{SO}(n)$. One particular case of that is the Normalized Essential manifold, corresponding to $k = 2$ and $n = 3$, which plays an important role in image processing. The classical problem of reconstructing a scene, or a video, from several images of the scene can be formulated as an interpolation problem on that manifold, since it encodes the epipolar constraint. Typically, it is given an ordered set of time-labeled essential matrices, $E_1, \ldots, E_j$, relating j different consecutive camera views (*snapshots*), and the aim is to calculate a continuum of additional virtual views by computing a smooth interpolating curve through the E_i's, $i = 1, \ldots, j$.

The De Casteljau algorithm has been used for a long time and on a widely variety of different areas. For a general introduction of the classical version of this algorithm and an overview of the historical evolution and existing works dealing with this concept we can mention, for instance, [6] and literature cited therein. Since most real life problems require the application of the De Casteljau algorithm in manifolds different from the Euclidean space $\mathbb{R}^m$, generalizations of this algorithm to other manifolds where performed, and there are known in the literature many references of works relating to the study of the algorithm, in other manifolds, and with different approaches. Just to name a few, see for instance, [1,4,11,12].

Although the existence in the literature of different methods for interpolating data in manifolds, the motivation to use, in this work, for the Essential manifold, the generalization of the classical De Casteljau algorithm for geodesically complete Riemannian manifolds, was based essentially in the fact that this algorithm is a powerful tool widely used and with a geometric and an algebraic feature that with recursive geodesic interpolation, allows to generate interpolating polynomial curves on manifolds.

In order not to be too exhaustive in this introductory section, more details concerned with this topic will appear in development sections.

2 Preliminaries and Notations

In this section, we provide preliminaries and introduce some notations that will be used throughout the paper. In this sense, we start to notice that the matrix groups are the most common examples of Lie groups which have greatest application in computer vision problems, robotics, engineering and control theory problems. These groups are all subgroups of the General Linear Group $\mathrm{GL}_n(\mathbb{R})$, the group of all real $n \times n$ invertible matrices, whose Lie algebra is $\mathfrak{gl}(n)$, the set of all $n \times n$ matrices with real entries, equipped with the commutator of matrices $[A, B] = AB - BA$ as the Lie bracket. The adjoint operator in $\mathfrak{gl}(n)$ is defined by $\mathrm{ad}_A B := [A, B]$ and, in the sequel, we consider $\mathfrak{gl}(n)$ equipped with the Euclidean inner product

$$\langle X, Y \rangle = \mathrm{tr}\big(X^\top Y\big),\ \ X, Y \in \mathfrak{gl}(n). \tag{1}$$

Given $X \in \mathfrak{gl}(n)$, the *matrix exponential of* X, denoted by e^X, is the $n \times n$ real matrix given by the sum of the following convergent power series $\mathrm{e}^X = \sum_{k=0}^{+\infty} \dfrac{X^k}{k!}$, where X^0 is defined to be the identity matrix I.

The vector space of $\mathfrak{gl}(n)$ consisting of all symmetric matrices is denoted by $\mathfrak{s}(n)$, while $\mathfrak{so}(n)$ denotes the Lie subalgebra of $\mathfrak{gl}(n)$ consisting of all skew-symmetric matrices. It is well-known that $\mathfrak{gl}(n) = \mathfrak{s}(n) \oplus \mathfrak{so}(n)$, is a decomposition of the Lie algebra $\mathfrak{gl}(n)$, and that $[\mathfrak{so}(n), \mathfrak{so}(n)] \subset \mathfrak{so}(n)$, $[\mathfrak{so}(n), \mathfrak{s}(n)] \subset \mathfrak{s}(n)$ and $[\mathfrak{s}(n), \mathfrak{s}(n)] \subset \mathfrak{so}(n)$. Also, $\mathfrak{so}(n)$ and $\mathfrak{s}(n)$ are orthogonal with respect to the inner product (1). The rotation group $\mathrm{SO}(n)$, having $\mathfrak{so}(n)$ as its Lie algebra, will also play a relevant role here.

It is important to mention that the logarithms of an invertible matrix B are the solutions of the matrix equation $\mathrm{e}^X = B$, and when B is real and doesn't have eigenvalues in the closed negative real line, i.e., when $\sigma(B) \cap \mathbb{R}_0^- = \varnothing$, where $\sigma(B)$ denotes the spectrum of B, there exists a unique real logarithm of B whose spectrum lies in the infinite horizontal strip $\{z \in \mathbb{C} \,:\, -\pi < Im(z) < \pi\}$ of the complex plane. In this work we will only consider this logarithm, usually called the *principal logarithm* of B and hereafter denoted by $\log B$. When B belongs to the rotation group $\mathrm{SO}(n)$, then $\log B$ belongs to its Lie algebra $\mathfrak{so}(n)$. Further, when $\|B - I\| < 1$, $\log B$ is uniquely defined by the following convergent power series: $\log B = \displaystyle\sum_{k=1}^{+\infty} (-1)^{k+1} \frac{(B-I)^k}{k}$. This power series defines the principal logarithm for matrices which are close to the identity matrix. However, for $\alpha \in [-1, 1]$, $\log(B^\alpha) = \alpha \log B$, so that, making $\alpha = 1/2^k$, with $k \in \mathbb{Z}$, one has $\log\Big(B^{\frac{1}{2^k}}\Big) = \frac{1}{2^k} \log B$. Further, since $\displaystyle\lim_{k \to +\infty} \Big(B^{\frac{1}{2^k}}\Big) = I$, the previous expression allows to compute $\log B$ even for matrices B which are not close to the identity. This procedure is called *inverse scaling and squaring method* and can be found, e.g., in [7]. More properties of these matrices functions are available, for instance, in [7,8], but we emphasize here the identity

$e^A D e^{-A} = e^{\mathrm{ad}_A} D = D + [A, D] + \dfrac{1}{2!}[A, [A, D]] + \cdots$, which plays an important role in this work.

3 The Geometry of the Essential Manifold

Let n be a positive integer and let k be a positive integer smaller than n. The Generalized Essential manifold $G_{k,n} \times \mathrm{SO}(n)$, briefly, Essential manifold, is the cartesian product of the real Grassmann manifold $G_{k,n}$, consisting of all k-dimensional (linear) subspaces of $\mathbb{R}^n$, and the rotation orthogonal group $\mathrm{SO}(n)$ of all orientation preserving rotational transformations in $\mathbb{R}^n$. Therefore, we have that

$$G_{k,n} \times \mathrm{SO}(n) = \{(S, R) : \; S \in G_{k,n}, \; R \in \mathrm{SO}(n)\}\,. \tag{2}$$

This manifold is a smooth compact and connected manifold of real dimension $k(n-k) + n(n-1)/2$ which can be seen as an embedded submanifold of $\mathfrak{s}(n) \times \mathbb{R}^{n \times n}$.

Each subspace of the Grassmann manifold $G_{k,n}$ can be associated to a unique operator of orthogonal projections onto itself, with respect to the Euclidean metric, and it is well-known that these operators (or, equivalently, its matrices, called projection matrices) are symmetric, idempotent, and have rank k. Therefore, in this work, similarly to [2,9], and [14] we use the following matrix representation of the Grassmann manifold

$$G_{k,n} := \big\{S \in \mathfrak{s}(n) \;:\; S^2 = S \text{ and } \mathrm{rank}(S) = k\big\}, \tag{3}$$

which is also a smooth compact connected manifold of real dimension $k(n-k)$, and an isospectral manifold, where each element has the eigenvalues 1 and 0, with multiplicity k and $n-k$, respectively. Defining, for an arbitrary point $S \in G_{k,n}$, the sets of matrices

$$\mathfrak{gl}_S(n) := \{A \in \mathfrak{gl}(n) \;:\; A = SA + AS\}; \quad \mathfrak{so}_S(n) := \mathfrak{so}(n) \cap \mathfrak{gl}_S(n), \tag{4}$$

the tangent space at a point $P = (S, R) \in G_{k,n} \times \mathrm{SO}(n)$ can be defined by

$$\begin{aligned} T_P\left(G_{k,n} \times \mathrm{SO}(n)\right) &= \{([\Omega, S], RY) \;:\; \Omega \in \mathfrak{so}_S(n),\, Y \in \mathfrak{so}(n)\} \\ &= \big\{\big(\mathrm{ad}_S^2(A), RY\big) \;:\; A \in \mathfrak{s}(n),\, Y \in \mathfrak{so}(n)\big\}\,. \end{aligned} \tag{5}$$

We consider the Essential manifold equipped with the Riemannian metric, induced by the Euclidean inner product on each tangent space, given by

$$\langle ([\Omega_1, S], RY_1)\,, ([\Omega_2, S], RY_2)\rangle = \mathrm{tr}(\Omega_1^T \Omega_2) + \mathrm{tr}(Y_1^T Y_2). \tag{6}$$

Moreover, using the last description of the tangent space at $P \in G_{k,n} \times \mathrm{SO}(n)$, the normal space at P, with respect to the Riemannian metric (6), can be defined as follows

$$\left(T_P\left(G_{k,n} \times \mathrm{SO}(n)\right)\right)^{\perp} = \big\{\big(A - \mathrm{ad}_S^2(A),\, RB\big) \;:\; A,\; B \in \mathfrak{s}(n)\big\}\,. \tag{7}$$

Bellow, we present a result for the Grassmann manifold $G_{k,n}$, that will be important to better understand further developments, and whose proof can be found in [13].

Lemma 1. *Let $S \in G_{k,n}$, $A, B \in \mathfrak{gl}_S(n)$ and $t \in \mathbb{R}$. Then,*

$$[A, S] = [B, S] \Leftrightarrow A = B; \tag{8}$$

$$\mathrm{e}^{2tA}(I - 2S) = \mathrm{e}^{\mathrm{ad}_{tA}}(I - 2S). \tag{9}$$

Now, we present some results about geodesics in the Essential manifold with respect to the Riemannian metric in (6).

3.1 Geodesics and Geodesic Distance

In this subsection, as mention, we present some results in order to compute explicitly, in $G_{k,n} \times \mathrm{SO}(n)$, the geodesic satisfying some initial conditions, the minimizing geodesic arc joining two points and the geodesic distance between two points of the Essential manifold.

Proposition 1. *The unique geodesic $t \mapsto \gamma(t)$ in $G_{k,n} \times \mathrm{SO}(n)$, satisfying the initial conditions $\gamma(0) = (S, R)$ and $\dot{\gamma}(0) = ([\Omega, S], RY)$, where $\Omega \in \mathfrak{so}_S(n)$ and $Y \in \mathfrak{so}(n)$, is given by*

$$\gamma(t) = (\gamma_1(t), \gamma_2(t)) = \left(\mathrm{e}^{t\Omega} S \mathrm{e}^{-t\Omega},\, R\mathrm{e}^{tY}\right) = \left(\mathrm{e}^{t\,\mathrm{ad}_\Omega} S,\, R\mathrm{e}^{tY}\right). \tag{10}$$

Proof. A detailed proof for the expression of γ_1 in $G_{k,n}$ can be found in [2], and since geodesics on $\mathrm{SO}(n)$ are translations of one parameter subgroups of $\mathrm{SO}(n)$ the proof of the expression γ_2 is immediate.

Remark 1. We note that any two points in the Essential manifold can be joined by a geodesic. This follows from the fact that $G_{k,n} \times \mathrm{SO}(n)$ is geodesically complete. However, to obtain an explicit formula for the geodesic that joins two points it is necessary to require some restrictions expressed in the following result.

Proposition 2. *Let $P = (S_1, R_1)$, $Q = (S_2, R_2) \in G_{k,n} \times \mathrm{SO}(n)$ be such that $\sigma\left((I - 2S_2)(I - 2S_1)\right) \cap \mathbb{R}^- = \varnothing$ and $\sigma(R_1^{-1}R_2) \cap \mathbb{R}^- = \varnothing$. Then, the minimizing geodesic arc in $G_{k,n} \times \mathrm{SO}(n)$, with respect to the Riemannian metric (6), that joins P (at $t = 0$) to Q (at $t = 1$), is parameterized explicitly by*

$$\gamma(t) = (\gamma_1(t), \gamma_2(t)) = \left(\mathrm{e}^{t\,\mathrm{ad}_\Omega} S_1,\, R_1\mathrm{e}^{tY}\right), \tag{11}$$

or, equivalently, by

$$\gamma(t) = (\gamma_1(t), \gamma_2(t)) = \left(\mathrm{e}^{t\,\mathrm{ad}_\Omega} S_1,\, \mathrm{e}^{t\overline{Y}} R_1\right), \tag{12}$$

where

$$\begin{aligned} &\Omega := \tfrac{1}{2}\log((I - 2S_2)(I - 2S_1)) \in \mathfrak{so}_{S_1}(n) \\ &Y := \log(R_1^{-1}R_2), \quad \overline{Y} := \log(R_2R_1^{-1}) \in \mathfrak{so}(n). \end{aligned} \tag{13}$$

Proof. The proof of the expression γ_2 is immediate. The proof of the expression γ_1, for the best of our knowledge, appeared for the first time in [2], and an easier alternative proof, taking in account Lemma 1, can be found in [13].

Remark 2. Notice that, since $\sigma(R_2R_1^{-1}) = \sigma(R_1^{-1}R_2)$, we have that the condition $\sigma(R_1^{-1}R_2) \cap \mathbb{R}^- = \varnothing$, automatically, implies that $\sigma(R_2R_1^{-1}) \cap \mathbb{R}^- = \varnothing$, and thus $\overline{Y}$ is well defined.

We end this section with the geodesic distance between two points P and Q in $G_{k,n} \times \mathrm{SO}(n)$, witch is the length of the minimal geodesic curve connecting them. Therefore, considering the Riemannian metric (6), it is possible to reach the next result, whose proof is immediate.

Proposition 3. *Let $P = (S_1, R_1), Q = (S_2, R_2) \in G_{k,n} \times \mathrm{SO}(n)$ be such that $\sigma((I - 2S_2)(I - 2S_1)) \cap \mathbb{R}^- = \varnothing$ and $\sigma(R_1^{-1}R_2) \cap \mathbb{R}^- = \varnothing$. Then, the geodesic distance between the points P and Q is given, explicitly, by*

$$\begin{aligned} d^2(P,Q) &= d_1^2(S_1,S_2) + d_2^2(R_1,R_2) \\ &= -\frac{1}{4}\mathrm{tr}\left(\log^2((I-2S_2)(I-2S_1))\right) - \mathrm{tr}\left(\log^2(R_1^{-1}R_2)\right) \end{aligned} \tag{14}$$

or, equivalently, by

$$d^2(P,Q) = -\frac{1}{4}\mathrm{tr}\left(\log^2((I-2S_2)(I-2S_1))\right) - \mathrm{tr}\left(\log^2(R_2R_1^{-1})\right). \tag{15}$$

4 De Casteljau Algorithm in the Essential Manifold

We start this section by presenting a review of the generalized De Casteljau algorithm to generate cubic polynomials in geodesically complete Riemannian manifolds. Then, following the approach of [4] for connected and compact Lie groups and for spheres, and the work of [14] for the Grassmann manifold we present a detailed implementation of the De Casteljau algorithm for the generation of cubic polynomials in the Essential manifold.

4.1 Review of the Generalized de Casteljau Algorithm

Let M be a m-dimensional connected Riemannian manifold, which is also geodesically complete, so that any pair of points may be joined by a geodesic arc.

The classical version of the De Casteljau algorithm was developed at 1959 by Paul De Casteljau [5], and it is a recursive process to generate interpolating polynomial curves of arbitrary degree in the Euclidean space $\mathbb{R}^m$. These curves, generated by successive linear interpolation, are known in the literature as Bernstein-Bézier curves or, simply, Bézier curves (see, e.g., [6]).

The generalized De Casteljau algorithm is an extension of this classical version for an arbitrary manifold, and where the linear interpolation is replaced by geodesic interpolation, this being the reason why we need to assume that the manifold M is geodesically complete, at least in a sufficiently big neighbourhood of the given data points.

In what follows, we present a description of this general De Casteljau algorithm for generating geometric cubic polynomials on M. For the sake of simplicity, we parameterize the curves on the $[0, 1]$ interval, but notice that there's no loss of generality, since this interval can be replaced by any other interval $[a, b]$, $a < b$, just by choosing the reparametrization $(t \to s)$ defined by $t = (s - a)/(b - a)$.

Therefore, we have that given a set of four distinct points $\{x_0, x_1, x_2, x_3\}$ in M, a smooth curve $t \in [0, 1] \mapsto \beta_3(t) := \beta_3(t, x_0, x_1, x_2, x_3)$ in M, joining x_0 (at $t = 0$) and x_3 (at $t = 1$), can be constructed by three successive geodesic interpolation steps as follows.

Generalized de Casteljau Algorithm

Step 1. Construct three geodesic arcs $\beta_1(t, x_i, x_{i+1})$, $t \in [0, 1]$ joining, for $i = 0, 1, 2$, the points x_i (at $t = 0$) and x_{i+1} (at $t = 1$).

Step 2. Construct two families of geodesic arcs

$$\beta_2(t, x_0, x_1, x_2) = \beta_1(t, \beta_1(t, x_0, x_1), \beta_1(t, x_1, x_2)),$$
$$\beta_2(t, x_1, x_2, x_3) = \beta_1(t, \beta_1(t, x_1, x_2), \beta_1(t, x_2, x_3)),$$

joining, for $i = 0, 1$ and $t \in [0, 1]$, the point $\beta_1(t, x_i, x_{i+1})$ (at $t = 0$) with the point $\beta_1(t, x_{i+1}, x_{i+2})$ (at $t = 1$).

Step 3. Construct the family of geodesic arcs

$$\beta_3(t, x_0, x_1, x_2, x_3) = \beta_1(t, \beta_2(t, x_0, x_1, x_2), \beta_2(t, x_1, x_2, x_3)),$$

joining, for each $t \in [0, 1]$, the points $\beta_2(t, x_0, x_1, x_2)$ (at $t = 0$) and $\beta_2(t, x_1, x_2, x_3)$ (at $t = 1$).

The curve $t \in [0, 1] \mapsto \beta_3(t) := \beta_3(t, x_0, x_1, x_2, x_3)$ obtained in Step 3. of the previous Algorithm is called *geometric cubic polynomial* in M. It is important to observe that this curve joins the points x_0 (at $t = 0$) and x_3 (at $t = 1$), but does not pass through the other two points x_1 and x_2. We illustrate this situation in Fig. 1, for the particular situation when the manifold M is the Euclidean space $\mathbb{R}^m$.

These points are usually called by *control points*, since they influence the shape of the curve. This algorithm can also be used to generate $\mathcal{C}^2$-smooth *geometric cubic polynomial splines* by piecing together, in a smooth manner, several geometric cubic polynomials and which are interpolating curves of the data.

A few remarks should be made concerning the general applicability of this construction. In fact, although the geometry of a Riemannian manifold possesses enough structure to formulate the construction, the basic ingredients used, the geodesic arcs, are implicitly defined by a set of nonlinear differential equations. Therefore, this algorithm can be only practically implemented when we can reduce the calculation of these geodesic arcs to a manageable form.

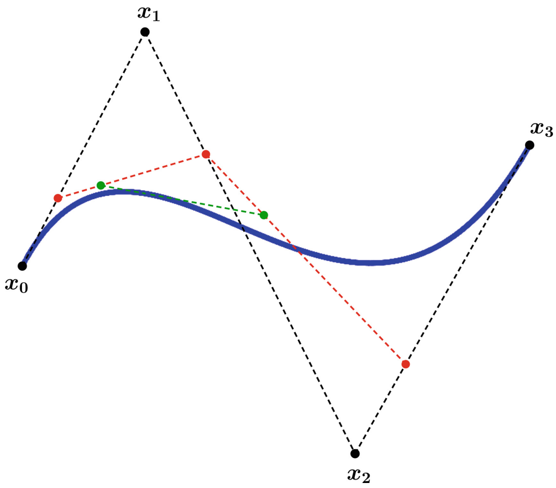

Fig. 1. Cubic polynomial defined by the De Casteljau algorithm in $\mathbb{R}^m$.

For this reason, in what follows we will present the implementation of the generalized De Casteljau algorithm, when the manifold M is the Essential manifold.

4.2 Implementation of the de Casteljau Algorithm in the Essential Manifold

Although the Essential manifold is geodesically complete, we have seen that an explicit formula for the geodesic that joins two points may be unknown in some particular situations. So, in this case the implementation of the De Casteljau algorithm is restricted to a convex open subset of the manifold where the expression to compute the geodesic arc joining two points is well-defined.

The generation of a geometric cubic polynomial in the Essential manifold $G_{k,n} \times \mathrm{SO}(n)$, starts from four given points $x_0 = (S_0, R_0)$, $x_1 = (S_1, R_1)$, $x_2 = (S_2, R_2)$, $x_3 = (S_3, R_3)$ in $G_{k,n} \times \mathrm{SO}(n)$.

Algorithm. Notice that, for $i = 0, 1, 2$ and $j = 1, 2, 3$, the superscripts in the operators Ω_i^j and V_i^j below are chosen according to the corresponding step.

Step 1. For $i = 0, 1, 2$, construct the geodesic arc joining the point x_i (at $t = 0$) with x_{i+1} (at $t = 1$), and given by

$$\beta_1(t, x_i, x_{i+1}) = (\beta_1(t, S_i, S_{i+1}),\ \beta_1(t, R_i, R_{i+1})) = \left(\mathrm{e}^{t\,\mathrm{ad}_{\Omega_i^1}} S_i,\ R_i \mathrm{e}^{tV_i^1}\right), \quad (16)$$

with

$$\begin{aligned} \Omega_i^1 &:= \frac{1}{2}\log((I - 2S_{i+1})(I - 2S_i)) \in \mathfrak{so}_{S_i}(n) \\ V_i^1 &:= \log(R_i^{-1} R_{i+1}) \in \mathfrak{so}(n). \end{aligned} \quad (17)$$

Step 2. Construct, for each $t \in [0,1]$, two geodesic arcs. The first, joining the points $\beta_1(t,x_0,x_1)$ and $\beta_1(t,x_1,x_2)$, and defined by

$$\beta_2(t,x_0,x_1,x_2) = \left(\mathrm{e}^{t\,\mathrm{ad}_{\Omega_0^2(t)}}\beta_1(t,S_0,S_1),\; R_0\mathrm{e}^{t\,V_0^1}\mathrm{e}^{tV_0^2(t)}\right), \tag{18}$$

with

$$\begin{aligned} \Omega_0^2(t) &:= \frac{1}{2}\log((I-2\beta_1(t,S_1,S_2))(I-2\beta_1(t,S_0,S_1))) \in \mathfrak{so}_{\beta_1(t,S_0,S_1)}(n) \\ V_0^2(t) &:= \log(\mathrm{e}^{(1-t)V_0^1}\mathrm{e}^{tV_1^1}) \in \mathfrak{so}(n). \end{aligned} \tag{19}$$

The second one, joining the points $\beta_1(t,x_1,x_2)$ and $\beta_1(t,x_2,x_3)$, and given by

$$\beta_2(t,x_1,x_2,x_3) = \left(\mathrm{e}^{t\,\mathrm{ad}_{\Omega_1^2(t)}}\beta_1(t,S_1,S_2),\; R_1\mathrm{e}^{tV_1^1}\mathrm{e}^{tV_1^2(t)}\right), \tag{20}$$

with

$$\begin{aligned} \Omega_1^2(t) &:= \frac{1}{2}\log\left((I-2\beta_1(t,S_2,S_3))(I-2\beta_1(t,S_1,S_2))\right) \in \mathfrak{so}_{\beta_1(t,S_1,S_2)}(n) \\ V_1^2(t) &:= \log(\mathrm{e}^{(1-t)V_1^1}\mathrm{e}^{tV_2^1}) \in \mathfrak{so}(n). \end{aligned} \tag{21}$$

Step 3. Construct, for each $t \in [0,1]$, the geodesic arc connecting the point $\beta_2(t,x_0,x_1,x_2)$ with the point $\beta_2(t,x_1,x_2,x_3)$, and defined by

$$\beta_3(t,x_0,x_1,x_2,x_3) = \left(\mathrm{e}^{t\,\mathrm{ad}_{\Omega_0^3(t)}}\beta_2(t,S_0,S_1,S_2),\; R_0\mathrm{e}^{tV_0^1}\mathrm{e}^{tV_0^2(t)}\mathrm{e}^{tV_0^3(t)}\right), \tag{22}$$

with

$$\begin{aligned} \Omega_0^3(t) &:= \frac{1}{2}\log((I-2\beta_2(t,S_1,S_2,S_3))(I-2\beta_2(t,S_0,S_1,S_2)))\in\mathfrak{so}_{\beta_2(t,S_0,S_1,S_2)}(n) \\ V_0^3(t) &:= \log(\mathrm{e}^{(1-t)V_0^2(t)}\mathrm{e}^{tV_1^2(t)}) \in \mathfrak{so}(n). \end{aligned} \tag{23}$$

As a result, and taking into consideration (16), (18), (20) and (22), we obtain the geometric cubic polynomial in the Essential manifold.

Definition 1. *The curve* $t \in [0,1] \mapsto \beta_3(t) := \beta_3(t,x_0,x_1,x_2,x_3)$ *defined by*

$$\beta_3(t) = \left(\mathrm{e}^{t\,\mathrm{ad}_{\Omega_0^3(t)}}\mathrm{e}^{t\,\mathrm{ad}_{\Omega_0^2(t)}}\mathrm{e}^{t\,\mathrm{ad}_{\Omega_0^1}}S_0,\; R_0\mathrm{e}^{tV_0^1}\mathrm{e}^{tV_0^2(t)}\mathrm{e}^{tV_0^3(t)}\right), \tag{24}$$

with Ω_0^1, Ω_0^2, Ω_0^3, V_0^1, V_0^2 *and* V_0^3 *given by (17), (19) and (23), is called a* geometric cubic polynomial *in the Essential manifold* $G_{k,n} \times \mathrm{SO}(n)$*, associated to the points* x_i*,* $i = 0,1,2,3$*.*

Remark 3. Notice that, as expected, the curve just defined joins the points x_0 (at $t=0$) and x_3 (at $t=1$). It is obvious that $\beta_3(0) = x_0$. To see that $\beta_3(1) = x_3$, we start to observe that, from the definition of Ω_i^j and V_i^j, $i = 0,1,2$, $j = 1,2,3$, it can be easily derived the following boundary conditions:

– $\Omega_0^2(0) = \Omega_0^3(0) = \Omega_0^1$, $\Omega_1^2(0) = \Omega_1^1$, $\Omega_1^2(1) = \Omega_0^3(1) = \Omega_2^1$, and $\Omega_0^2(1) = \Omega_1^1$;
– $V_0^2(0) = V_0^3(0) = V_0^1$, $V_1^2(0) = V_1^1$, $V_1^2(1) = V_0^3(1) = V_2^1$, and $V_0^2(1) = V_1^1$.

Thus, considering these boundary conditions together with the definition of the geodesic arcs (16), we obtain that $\beta_3(1) = x_3$.

5 Conclusion and Future Work

We have presented all the details to implement the De Casteljau algorithm in the Essential manifold. As regarded, the implementation of the algorithm in this manifold requires that one knows how to compute matrix exponentials of skewsymmetric matrices and logarithms of orthogonal matrices.

For practical applications in data mining, one still has to rely on computing stable matrix exponentials and logarithms of structured matrices, since there are no explicit formulas to compute those matrix functions. However, when $k = 2$ and $n = 3$, we have explicit formulas to compute those matrices functions.

Efficient numerical methods to compute matrix functions have been developed along the years (see, for instance, [7]), and this is one of our future work.

Acknowledgements. The authors acknowledge Fundação para a Ciência e a Tecnologia (FCT) and COMPETE 2020 program for financial support to projects UIDB/00048/2020 and UIDB/04466/2020.

References

1. Altafini, C.: The De Casteljau algorithm on SE(3). In: Lecture Notes in Control and Information Sciences. Springer, London (2007). https://doi.org/10.1007/BFb0110205
2. Batzies, E., Hüper, K., Machado, L., Silva Leite, F.: Geometric mean and geodesic regression on grassmannians. Linear Algebra Appl. **466**, 83–101 (2015)
3. Bressan, B.: From Physics to Daily Life: Applications in Informatics, Energy, and Environment. Wiley-Blackwell, Hoboken (2014)
4. Crouch, P., Kun, G., Leite, F.S.: The De Casteljau algorithm on Lie groups and spheres. J. Dyn. Control Syst. **5**(3), 397–429 (1999)
5. de Casteljau, P.: Outillages Méthodes Calcul. Technical Report, André Citroën Automobiles SA (1959)
6. Farin, G.: Curves and Surfaces for Computer-Aided Geometric Design - A Practical Guide, 5th edn. Academic Press, New York (2002)
7. Higham, N.: Functions of Matrices: Theory and Computation. SIAM (2008)
8. Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)
9. Hüper, K., Leite, F.S.: On the geometry of rolling and interpolation curves on S^n, SO(n), and Grassmann manifolds. J. Dyn. Control Syst. **4**, 467–502 (2007)
10. Ma, Y., Soatto, S., Kosecká, J., Sastry, S.: An Invitation to 3D Vision: From Images to Geometric Models. Springer, New York (2004). https://doi.org/10.1007/978-0-387-21779-6
11. Nava-Yazdani, E., Polthier, K.: De Casteljau's algorithm on manifolds. Comput. Aided Geom. Des. **30**(7), 722–732 (2013)

12. Park, F., Ravani, B.: Bézier curves on Riemannian manifolds and Lie groups with kinematics applications. ASME J. Mech. Des. **117**, 36–40 (1995)
13. Pina, F., Silva Leite, F.: Cubic splines in the Grassmann manifold generated by the De Casteljau algorithm. In: Preprint Series - Department of Mathematics - University of Coimbra, Portugal, no. 20–07 (2020)
14. Pina, F., Silva Leite, F.: Cubic splines in the Grassmann manifold. In: Gonçalves, J.A., Braz-César, M., Coelho, J.P. (eds.) CONTROLO 2020. LNEE, vol. 695, pp. 243–252. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-58653-9_23
15. Szeliski, R.: Computer Vision: Algorithms and Applications. Texts in Computer Science, Springer, London (2011). https://doi.org/10.1007/978-1-84882-935-0