# Exploring quasi-geodesics on Stiefel manifolds in order to smooth interpolate between domains

Jorge Batista* Krzysztof Krakowski** Fátima Silva Leite***

***Abstract*— Manifold-based algorithms are receiving increasing attention in computer vision and pattern recognition. Geodesic curves in the Graßmann manifold have proven to be very useful in modeling domain shift between a source and target domain, represented as subspaces. To obtain an invariant domain representation, the data is projected into a set of subspaces along the geodesic.**

**In contrast to previous works that mainly explore intermediate subspaces along geodesics, in this paper we propose an alternative approach to address multiple source domain adaptation, by taking advantage of smooth interpolating curves on the Stiefel manifold to walk along a set of multiple domains. This aspect is particularly interesting in temporally or dynamically evolving events that are represented by discrete subsets of the data.**

**To generate such curves, we apply a recent technique based on successive quasi-geodesic interpolation on the Stiefel manifold, that results from a modification of the Casteljau algorithm.**

**To evaluate the usefulness of these smooth interpolating curves in pattern recognition problems, several experiments were conducted. We show the advantage of using such curves in multi-source unsupervised domain adaptation problems and in object recognition problems across dynamically evolving datasets.**

## I. Introduction

Stiefel and Graßmann manifolds arise naturally in computer vision applications and pattern recognition, since features and patterns that describe visual objects may be represented as elements in those manifolds. These geometric representations facilitate the analysis of the underlying geometry of the data. In this paper we define the Graßmann manifold as the set of all $k$-dimensional subspaces in $\mathbb{R}^n$ and the Stiefel manifold as the set of all $k$ orthonormal vectors in $\mathbb{R}^n$.

Graßmann and Stiefel manifolds are clearly related, but the former is a symmetric space which makes its geometry less complicated than the geometry of the latter. This reflects even on solutions of simple formulated problems, such as that of finding explicit formulas for geodesics that join two given points, which in turn is a basic step to solve other important problems, namely averaging, fitting and smoothing. A formula for the geodesic that joins two points on Graßmann manifolds and depends explicitly only on those points can be found in [1]. But for Stiefel manifolds, even the simpler problem of finding a geodesic that starts at a given point with a prescribed velocity is not so straightforward, as can be seen for instance in the work of [2].

Geodesic curves on Graßmann manifold have proven to be very useful in modeling domain shift between a source and target domain represented as subspaces. In contrast, multiple source domain shifts on manifold has received much less attention, in particular those related with temporally or dinamically evolution. The majority of the multiple source domain adaptation solutions that use subspace representations to model the domains are based on leveraging or selecting source domains, making use of a single subspace to represent the multiple source domains. This single subspace can be obtained through a Karcher mean computation or by selecting the most representative domains, either by means of a rank of domains or by learning optimal weights for different source domains.

In the present paper we explore a recent technique to generate a smooth curve on a Stiefel manifold that interpolates a given set of data frames. This provides an alternative approach to address multiple source domain adaptation, where a smooth interpolation curve on the Stiefel manifold is used to walk along a set of multiple domains.

The interpolating smooth curve on the Stiefel manifold will be generated by a recent geometric algorithm presented in [3] which results from a modification of the Casteljau algorithm on manifolds in which geodesics are replaced by quasi-geodesics, simple curves that can be defined explicitly. This algorithm is intrinsic to the manifold since the whole geometric construction, which is based on successive quasi-geodesic interpolation. lives on the Stiefel manifold.

**Main contribution**: We present a new approach, based on the Stiefel manifold of $k$-frames in $\mathbb{R}^n$, to address the multiple source domain adaptation problem. We demonstrate the applicability of the interpolation algorithm on face recognition across expressions and illumination variations, cross dataset object recognition, and report the improved performance of our approach over existing multiple-source unsupervised domain adaptation methods.

## II. Related Work

Domain adaptation is a fundamental problem in machine learning that has attracted the attention of a multitude of related research fields. A nice survey of the recent advances in visual domain adaptation can be found in [4], [5]. For the purpose of this paper, we briefly review what we consider the more relevant work to our methodology.

*Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal, and Department of Electrical Engineering and Computers, University of Coimbra, 3030-290 Coimbra, Portugal.

**Wydział Matematyczno-Przyrodniczy, Uniwersytet Kardynała Stefana Wyszyńskiego, Warsaw, Poland, and University of Melbourne, Australia.

***Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal, and Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal.

Geodesic curves on a Graßmann manifold have proven to be very useful in modeling domain shift between a source and a target domain represented as subspaces. A pioneer work oriented to unsupervised domain adaptation on manifolds was presented in [6]. Instead of considering just the information provided by the source and target domains, [6] uses incremental learning by finite sampling intermediate subspaces along the geodesic curve connecting the source and target subspaces on the Graßmann manifold. That paper was the forerunner of other domain adaptation solutions that explored the idea of interpolation on manifolds (see [5]).

To avoid the ad-hoc sampling of intermediate subspaces used in [6], kernelized solutions were proposed that extended the sampling problem to the infinite case, defining a new kernel equivalent to integrating over all common subspaces that lie on the geodesic flow connecting the source and target subspaces [7], [8], [9]. Supported on this kernelized solution, [10] proposed a solution based on the parallel transport of union of the source subspaces on the Graßmann manifold.

The idea of interpolating subspaces through dictionary learning to link the source and the target domain was presented in [11]. This unsupervised domain adaptive dictionary learning solution allows the synthesis of data associated with the intermediate domains while exploiting the discriminative power of generative dictionaries. Recognition under domain shifts is accomplished using a classifier built with synthetic data obtained from the intermediate domains. This idea of dictionary learning was used to model dynamically evolving events, which was also explored in [12], [13], [14], and [15] on their domains adaptation approaches oriented to continuous temporal evolution of the target domain.

More recently, [16] extended the concept of interpolating between subspaces to the deep learning paradigm. Under this paradigm, the DLIB solution is able to learn a hierarchical representation of the data while trying to take into account domain shift.

The idea of interpolating domains has also been explored in [17] by means of shifting covariance. By representing the domains as covariance matrices, intermediate domains are interpolated along geodesics on the SPD manifold to model the domain shift between the domains.

Multiple source domain shifts on manifold has also called the attention of the community[18], [19], [20], [21], [22], [8], [9], [17], [23]. The problem of multi-source domain adaptation on manifolds was addressed in [24], by exploring the concept of rolling to generate a $C^1$-smooth interpolating curve on the Graßmann manifold. This approach uses rolling to project the data from the manifold to a vector space, then interpolation on this simple space is performed, and finally the resulting curve is mapped back to the manifold.

Contrary to the idea of rolling on manifolds, [23] explored the idea of sampling subspaces along smooth curves on Graßmann by computing interpolating curves intrinsically on the manifold. The idea of shifting covariance [17] has also been extended to the problem of multi-source domain adaptation by computing each geodesic curve that connects a source domain to a target domain and jointly learning multiple classifiers and optimal weights for each source domains.

What we propose on this paper is to compute interpolation curves intrinsically on the Stiefel manifold, using a convenient modification of the Casteljau algorithm, which consists on replacing successive geodesic interpolation by successive quasi-geodesic interpolation. This overcomes the difficulties that arise from not knowing explicit formulas for geodesics that join two arbitrary points on the Stiefel manifold. Fitting smoothing curves on certain manifolds using only a set of indexed samples was also addressed in [25].

By interpolating along smooth curves on manifolds, our algorithm is able to correctly model the domain shifts between temporally or dynamically evolving domains.

## III. Revisiting Stiefel manifolds

We include here basic facts about Stiefel manifolds that can be found in [2] and also refer to [3] for further details.

Let $\mathfrak{s}(n)$ and $\mathfrak{so}(n)$ denote respectively the set of all $n \times n$ real symmetric matrices and the set of all $n \times n$ real skew-symmetric matrices. For $P \in \mathfrak{s}(n)$, define $\mathfrak{so}_P(n) := \{\, X \in \mathfrak{so}(n) : XP + PX = X \,\}$.

The Stiefel manifold, hereafter denoted by $\mathscr{S}_{n,k}$, is the set of orthonormal $k$-frames in $\mathbb{R}^n$. In matrix representation

$$\mathscr{S}_{n,k} := \left\{\, S \in \mathbb{R}^{n\times k} \;:\; S^{\mathrm{T}}S = I_k \,\right\}.$$

The tangent space to $\mathscr{S}_{n,k}$ at a point $S \in \mathscr{S}_{n,k}$ is usually defined as $T_S\mathscr{S}_{n,k} = \left\{\, V \in \mathbb{R}^{n\times k} \;:\; V^{\mathrm{T}}S + S^{\mathrm{T}}V = 0 \,\right\}$, but it can also be parametrized as

$$T_S\mathscr{S}_{n,k} = \{\, V = XS + S\Omega : X \in \mathfrak{so}_P(n), \Omega \in \mathfrak{so}(k) \,\}, \quad (1)$$

where $P = SS^{\mathrm{T}}$. Moreover, if $V \in T_S\mathscr{S}_{n,k}$ is known, one can obtain $X$ and $\Omega$ in (1) since, as proved in [3],

$$X = VS^{\mathrm{T}} - SV^{\mathrm{T}} + 2SV^{\mathrm{T}}SS^{\mathrm{T}} \quad \text{and} \quad \Omega = S^{\mathrm{T}}V. \quad (2)$$

The Stiefel manifold is considered equipped with the *canonical metric* defined by

$$\langle V_1, V_2\rangle = \mathrm{tr}\big(V_1^{\mathrm{T}}(I - \tfrac{1}{2}SS^{\mathrm{T}})V_2\big), \quad \text{for } V_1, V_2 \in T_S\mathscr{S}_{n,k}.$$

We are interested in finding explicit formulas for simple curves that join two points in the Stiefel manifold. Geodesics would be the natural candidates, but it turns out that, except for some particular cases, explicit formulas for the geodesic that joins two arbitrary points are not known. We need some extra facts before introducing those alternative curves.

Let $\mathbb{SO}(n)$ denote the group of special orthogonal $n \times n$ matrices. $\mathbb{SO}(n) \times \mathbb{SO}(k)$ acts transitively on $\mathscr{S}_{n,k}$ via

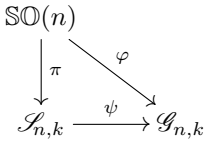
$$\begin{array}{ccc} \mathbb{SO}(n)\times\mathbb{SO}(k)\times\mathscr{S}_{n,k} & \longrightarrow & \mathscr{S}_{n,k} \\ (\Theta_1, \Theta_2, S) & \mapsto & \Theta_1 S \Theta_2. \end{array}$$

When $X \in \mathfrak{so}(n)$ and $\Omega \in \mathfrak{so}(k)$, $t \mapsto e^{tX}$ and $t \mapsto e^{t\Omega}$ are geodesics in $\mathbb{SO}(n)$ and $\mathbb{SO}(k)$ respectively, passing through the identity at $t = 0$, but the curve $t \mapsto \beta(t) = e^{tX}\, S\, e^{t\Omega}$ in $\mathscr{S}_{n,k}$ is not always a geodesic w.r.t. the canonical metric. Nevertheless, these curves, called *quasi-geodesics* in [3], have very interesting properties that are summarized in the

theorem below, where $D_t\dot{\gamma}$ denotes covariant acceleration along a curve $\gamma$ and $\log(Y)$ is the principal logarithm of a nonsingular matrix Y, which always exists and is unique as long as Y has no negative eigenvalues ([26]). We recall the relationship between the rotation group $\mathbb{SO}(n)$, the Stiefel manifold $\mathscr{S}_{n,k}$ and the Graßmann manifold $\mathscr{G}_{n,k}$ of all $k$-dimensional linear subspaces in $\mathbb{R}^n$, which has the following matrix representation

$$\mathscr{G}_{n,k} := \left\{ P \in \mathfrak{s}(n) \ : \ P^2 = P \text{ and } \operatorname{rank}(P) = k \right\}.$$

Let $\Delta := \begin{bmatrix} \mathbf{I}_k \\ 0 \end{bmatrix}_{n\times k} \in \mathscr{S}_{n,k}$, and $\Lambda := \Delta\Delta^{\mathrm{T}} \in \mathscr{G}_{n,k}$. The following commutative diagram, where $\pi(\Theta) := \Theta\Delta$, $\varphi(\Theta) := \Theta\Lambda\Theta^{\mathrm{T}}$, and $\psi(S) := SS^{\mathrm{T}}$,




defines relationships between $\mathbb{SO}(n)$, $\mathscr{G}_{n,k}$ and $\mathscr{S}_{n,k}$. In particular, the projection map $\psi$ sends a $k$-frame $S$ into the subspace it generates, hereafter represented by $P = SS^{\mathrm{T}}$. Moreover, this mapping is not one-to-one since for $P \in \mathscr{G}_{n,k}$,

$$\psi^{-1}(P) = \left\{ S\,\mathbb{SO}(k) : \ S \in \mathscr{S}_{n,k} \text{ and } SS^{\mathrm{T}} = P \right\}.$$

This set is called the *fiber over the point* $S$.

*Theorem 1 ([3]):* Let $S_1$ and $S_2$ be two distinct points in $\mathscr{S}_{n,k}$. Then, if

$$X = \tfrac{1}{2}\log\big((I - 2S_2S_2^{\mathrm{T}})(I - 2S_1S_1^{\mathrm{T}})\big)$$

$$\text{and} \quad \Omega = \log\big(S_1^{\mathrm{T}}e^{-X}S_2\big),$$

the quasi-geodesic defined by

$$\beta(t) := e^{tX}S_1e^{t\Omega},$$

satisfies the following properties:

1) $\beta(0) = S_1 \quad \text{and} \quad \beta(1) = S_2$;
2) $\|\dot{\beta}(t)\|^2 = -\operatorname{tr}\big(S_1^{\mathrm{T}}X^2S_1 + \frac{1}{2}\Omega^2\big)$ (constant speed);
3) $D_t\dot{\beta}(t) = X\beta(t)\Omega \neq 0$, unless $X = 0$ or $\Omega = 0$;
4) When $X = 0$, the curve $\beta(t) := S_1e^{t\Omega}$ is a geodesic living in the fiber over $S_1$;
5) When $\Omega = 0$, the curve $\beta(t) := e^{tX}S_1$ is a geodesic;
6) $\|D_t\dot{\beta}(t)\|^2 = \operatorname{tr}\big(\Omega^2S_1^{\mathrm{T}}X^2S_1\big)$ (constant covariant acceleration).
7) $\gamma(t) = \beta(t)\beta(t)^{\mathrm{T}} = e^{tX}P_1e^{-tX}$ is the geodesic in $\mathscr{G}_{n,k}$ joining the points $P_1 = S_1S_1^{\mathrm{T}}$ and $P_2 = S_2S_2^{\mathrm{T}}$.

The scheme in the next figure illustrates the relationship between geodesics and quasi-geodesics in Stiefel, and geodesics in Graßmann. In particular, if the frames $S_1$ and $S_2$ generate the same subspace, they can be joined by a geodesic living in the fiber over $S_1$. If the frames $S_1$ and $S_2$ do not generate the same subspace, but $S_2 = e^XS_1$, they can also be joined by a geodesic. Otherwise, $S_1$ and $S_2$ can be joined by a broken geodesic (two pieces of geodesics) or by a smooth curve, a quasi-geodesic. There are some similarities between this scheme and Fig. 1 in [27], but the quasi-geodesic doesn't appear in the latter.

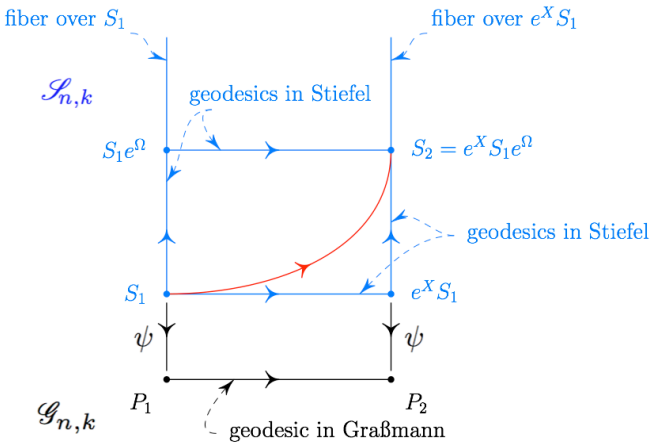



Fig. 1. Scheme showing, in red, the quasi-geodesic that joins points $S_1$ and $S_2$ in $\mathscr{S}_{n,k}$ and showing, in black, the geodesic in $\mathscr{G}_{n,k}$ which is the projection of the quasi-geodesic.

## IV. Solving an interpolation problem on the Stiefel manifold $\mathscr{S}_{n,k}$

### A. Formulation of the problem

> *Problem 1:* Given a set of points $\{S_i\}_{i=0}^m$ belonging to the Stiefel manifold $\mathscr{S}_{n,k}$, and a vector $V_0 \in T_{S_0}\mathscr{S}_{n,k}$, find a $\mathcal{C}^1$-smooth interpolating curve passing through these points and having initial velocity equal to $V_0$.

This is a particular interpolation problem on a Riemannian manifold that can be solved using the generalization of the Casteljau algorithm in [28] if explicit formulas for the geodesic that joins two points on the manifold are available. Generalizations of the Casteljau algorithm to manifolds appeared more recently in [29], [30], [31] and [32]. Recently, in [3], a modification of the Casteljau algorithm for the Stiefel manifold was proposed, where geodesic interpolation is replaced by quasi-geodesic interpolation. This is the approach taken here to solve Problem 1.

*Remark 2:* At first glance, one may think that an efficient method to solve the interpolation Problem 1 is to project the data to the Graßmann manifold via the projection mapping $\phi$, then implement the Casteljau algorithm on this manifold since explicit formulas for the geodesic that joins two points in the Graßmann manifold are known (such formulas have been derived in [1]), and finally projecting the interpolating solution curve back on the Stiefel manifold. However, since the projection mapping is not one-to-one, the smoothness conditions in the final curve would be almost impossible to meet.

### B. Solving the interpolation problem

The interpolating curve is constructed segment by segment, each segment joining two consecutive data points, $S_i$

and $S_{i+1}$, results from a two-step procedure using quasi-geodesics. An auxiliary point $C_i$ is required to build each segment. This point is called a *control point* since its choice affects the shape of the curve. The first control point is related to the prescribed initial velocity $V_0$, and the other control points are chosen in order that each segment joins its neighbor segments smoothly. Without loss of generality, we assume that all segments are parameterized in the $[0, 1]$ time interval.

*1) The basic two-step procedure on a manifold:* Given a set of three points $x_0, x_1, x_2$ in a manifold $\mathbf{M}$, let $t \mapsto \sigma_1(t, x_0, x_1)$ and $t \mapsto \sigma_1(t, x_1, x_2)$ be two smooth curves joining $x_0$ to $x_1$ and $x_1$ to $x_2$ respectively. Define a family of curves

$$\gamma \colon [0, 1] \times [0, 1] \to \mathbf{M}$$

as follows: For a fixed $\tau \in [0, 1]$, consider the curve $t \mapsto \gamma(t, \tau)$ joining the point $\sigma_1(\tau, x_0, x_1)$ to the point $\sigma_1(\tau, x_1, x_2)$. Then, $t \mapsto \sigma_2(t) = \gamma(t, t)$ is a smooth curve in $\mathbf{M}$ joining $x_0$ to $x_2$, where $x_1$ is the control point.

It turns out that the initial and final velocities of the curve $\sigma_2$ are related to the initial and final velocity of the curves $\sigma_1$, according to the following:

$$\dot\sigma_2(0) = 2\dot\sigma_1(0, x_0, x_1) \quad \text{and} \quad \dot\sigma_2(1) = 2\dot\sigma_1(1, x_1, x_2).$$

These identities establish a relationship between the initial and final velocity of the curve $\sigma_2$ and the control point $x_1$. In particular, when $\mathbf{M} = \mathscr{S}_{n,k}$ and $t \mapsto \sigma_1(t, x_i, x_{i+1})$, $i = 0, 1$, are quasi-geodesics, the control point $x_1$ can be obtained from the initial point and the initial velocity, both prescribed in Problem 1.

*2) Generating the first curve segment in $\mathscr{S}_{n,k}$, joining $S_0$ to $S_1$:* Following the basic two-step procedure, we first need to find a control point $C_0$, which is the end point of the quasi-geodesic that starts at the point $S_0$ with initial velocity equal to $\frac{1}{2}V_0$. This quasi-geodesic is given by

$$\beta_0(t) = e^{tX_0} S_0 e^{t\Omega_0},$$

where, according to (2),

$$X_0 = \frac{1}{2}V_0 S_0^{\mathrm{T}} - \frac{1}{2}S_0 V_0^{\mathrm{T}} + S_0 V_0^{\mathrm{T}} S_0 S_0^{\mathrm{T}}, \quad \Omega_0 = \frac{1}{2}S_0^{\mathrm{T}} V_0.$$

So, $C_0 = \beta_0(1) = e^{X_0} S_0 e^{\Omega_0}$ defines the control point.

We next proceed to the construction of the second quasi-geodesic $\beta_1$ that joins $C_0$ to $S_1$, using Theorem 1 with the obvious adaptations. The first curve segment, joining $S_0$ to $S_1$ with prescribed initial velocity equal to $V_0$ can now be obtained from quasi-geodesic interpolation of $\beta_0$ and $\beta_1$.

*3) Generating consecutive segments:* The second curve segment joins $S_1$ to $S_2$ and must be $\mathcal{C}^1$-smooth at $S_1$, which means that the initial velocity for this second curve segment must equal the end velocity of the previous curve segment, which is already known. So, we are reduced to the generation of a curve segment that joins $S_1$ to $S_2$ and whose initial velocity at $S_1$ is equal to $\dot\sigma(1)$. This data is enough to find a new control point $C_1$ which is necessary to generate the second curve segment.

The other consecutive segments are generated similarly. The procedure to generate the interpolating curve is summarized in algorithms 1 and 2.

**Algorithm 1:** Calculate a set of $K$ intermediate points $\sigma(t)$ in $\mathscr{S}_{n,k}$, for $t \in [0, 1]$, such that: $\sigma(0) = S$, $\sigma(1) = Q$

**Input**: $K$, $t \in [0, 1]$, $S, C, Q \in \mathscr{S}_{n,k}$, $V_0 \in T_S\mathscr{S}_{n,k}$, where $S$ is the initial point, $Q$ is the final point and $C$ is the control point
**Output**: $\sigma(t)$, $X_1 \in \mathfrak{so}_{CC^{\mathrm{T}}}(n)$, $\Omega_1 \in \mathfrak{so}(k)$

Calculate velocity components $X_0$ and $\Omega_0$:

$$X_0 = \frac{1}{2}\log\big((\mathbf{I} - 2CC^{\mathrm{T}})\,(\mathbf{I} - 2SS^{\mathrm{T}})\big)$$
$$\Omega_0 = \log(S^{\mathrm{T}} \exp(-X_0)\, C)$$

Calculate velocity components $X_1$ and $\Omega_1$:

$$X_1 = \frac{1}{2}\log\big((\mathbf{I} - 2QQ^{\mathrm{T}})\,(\mathbf{I} - 2CC^{\mathrm{T}})\big)$$
$$\Omega_1 = \log(C^{\mathrm{T}} \exp(-X_1)\, Q)$$

**for** $k \leftarrow 0$ **to** $K$ **do**
  $t = k/K$
  Calculate quasi-geodesics $\beta_0(t)$ and $\beta_1(t)$:

$$\beta_0(t) = \exp(tX_0)\, S\, \exp(t\Omega_0)$$
$$\beta_1(t) = \exp(tX_1)\, C\, \exp(t\Omega_1)$$

  Calculate velocity components $X(t)$ and $\Omega(t)$ for the joining segment:

$$X(t) = \frac{1}{2}\log\Big(\big(\mathbf{I} - 2\,\beta_1(t)\,\beta_1^{\mathrm{T}}(t)\big)\big(\mathbf{I} - 2\,\beta_0(t)\,\beta_0^{\mathrm{T}}(t)\big)\Big)$$
$$\Omega(t) = \log\Big(\beta_0^{\mathrm{T}}(t)\, \exp(-X(t))\, \beta_1(t)\Big)$$

  Compute the point on the spline segment:

$$\sigma(t) = \exp(tX(t))\, \beta_0(t)\, \exp(t\Omega(t))$$

**return** $\sigma$, $X_1$, $\Omega_1$.

**Algorithm 2:** Calculate control point $C$ given the initial point for the segment and velocity components of the previous segment

**Input**: $Q \in \mathscr{S}_{n,k}$, $X_1 \in \mathfrak{so}_{QQ^{\mathrm{T}}}(n)$ and $\Omega_1 \in \mathfrak{so}(k)$
**Output**: $C \in \mathscr{S}_{n,k}$

Calculate the control point $C$:

$$C = \exp(X_1)\, Q\, \exp(\Omega_1)$$

**return** $C$ .

## V. Domain Adaptation

Consider $D$ domains and a feature space of dimension $n$. In each domain $d = 1, \ldots, D$ take $m_d$ samples, each one represented by a column vector $x_i^d \in \mathbb{R}^n$, $(i = 1, \ldots, m_d)$. Such vectors form the columns of a matrix $X^d \in \mathbb{R}^{n \times m_d}$ which is the feature representation of the samples in domain $d$.

To establish a common ground for comparison, we follow very closely the protocol proposed in [6]. Specifically, orthogonal subspace estimation is applied to all samples in each domain to produce $D$ matrices, $\Theta^d$, in $\mathbb{SO}(n)$. Projecting each $\Theta^d$ on the Stiefel manifold $\mathscr{S}_{n,k}$, via the projection map $\pi$, one obtains $D$ matrices $S^d = \pi(\Theta^d) = \Theta^d\Delta$, whose columns form an orthonormal bases for the $k$-dimensional subspaces $P^d = \psi(S^d) = S^d(S^d)^{\mathrm{T}}$, which are points in the Graßmann manifold $\mathscr{G}_{n,k}$. Following Algorithm 3, which builds from Algorithm 1, we obtain a smooth curve on the manifold that interpolates the multiple domains represented

by the frames $S^d \in \mathscr{S}_{n,k}$. By sampling the interpolation curve at $J$ different values of the parameter $t$, we collect $J$ intermediate frames $S_j$, $j = 1, \ldots, J$ that generate new subspaces. Then, each sample from the multiple domains is projected onto these new subspaces to obtain $J$ projections $\hat{X}_j := (S_j)^{\mathrm{T}} X_i$, $j = 1, \ldots, J$, which are then concatenated to form a vector in $\mathbb{R}^{Jk}$. A discriminative classifier can be trained to classify unlabeled samples of a target domain, based on the high-dimensional feature vectors obtained using the labeled samples from source domains.

**Algorithm 3:** Computing intermediate frames by sampling smooth interpolation curves

**Input**: Multiple domain data $X^d$, subspace dimension $k$, number of intermediate samples per curve segment $K$
**Output**: Sampled points $S(j), j = 1, \ldots, J$, on $\mathscr{S}_{n,k}$. Each point represents a basis of vectors used to specify a subspace.

Perform PCA on each $X^d$ to obtain the orthonormal $k-$frame in $\mathbb{R}^n$, $S^d$.
Perform ordering (indexation) of $S^d$.
Consider $S^2$ as the control point for 1st curve segment.
Get $K$ sample points and velocity components (first segment).

$$[S, X, \Omega] = Alg_1(S^1, S^2, S^3)$$

**for** $i \leftarrow 3$ **to** $D - 1$ **do**
  Get control point for next segment

$$[C] = Alg_2(S^i, X, \Omega)$$

  Get $K$ sample points and velocity components (next segment).

$$[S^{'}, X, \Omega] = Alg_1(S^i, C, S^{i+1})$$

  Concatenate $[S] \leftarrow [S, S^{'}]$
**return** $S$

## VI. Experiments

To evaluate our solution, we conducted experiments on cross-domain object recognition and face recognition across expression and illumination variation. We compared our results with some well established algorithms from the state-of-art, the Geodesic Flow Kernel (GFK) algorithm [8], [9] and the Karcher-Sampling Geodesic Flow (K-SGF) algorithm [6], [22].

### A. Object Recognition Across Datasets

For a quantitative evaluation, we followed the protocol described in [8], [6], [22] using the OFFICE+Caltech10 benchmark dataset from [33] that contains four image datasets: Caltech, Amazon, DSLR and Webcam.


Fig. 2. Example images from the OFFICE+Caltech10 dataset [8].

The domain shift in this dataset is due to changes in object pose, image resolution, background clutters and scene illumination. Instead of using the SURF based image representation as in [33], [8], we used the VGG-Net features, in particular the VGG-FC7 features, that were recently extracted with the network model of [34] and also used in [35].

In all experiments conducted, sets of three source domains were considered and only samples from the source domains were labeled. The same 10 common categories reported in [8] were considered for evaluation. As discriminative classifier a simple 1-KNN classifier was used.

Instead of using PCA to define the subspaces, supervised PLS and OLPP [36] were used for that purpose. Subspace dimension and the number of intermediate subspaces were selected empirically and their values are presented in the performance tables.

For our results, accuracy was obtained by averaging over 20 experiment trials; each trial containing a random set of labeled samples from each one of the source domains. We report performance on different sources-target combinations. We also evaluated the performance when intermediate subspaces are sampled along non-smooth curves, broken geodesics (B-Geod), obtained by simply concatenating the geodesic curves $\sigma_1(t, x_i, x_{i+1})$. Results are shown in Table I. For comparative evaluation we include also several results reported in [35] using the same VGG-FC7 features. For each target domain we show the mean average result obtained from all possible indexation of source domains and also the best result.

Multi-source domain adaptation consistently achieved good results being the $\mathcal{C}^1$-smooth interpolation ranked among top performers, which empirically proves the advantage of our solution. Only in cases where extremely good adaptability between a particular pair of source and target domains exists, our solution does not reach top performance. This happened when image datasets, like Webcam and DSLR, were combined as source-target pair. In these favorable conditions (note the similarity in appearance between datasets Webcam and DSLR in Fig. 3), the incorporation of additional information coming from less similar datasets does not increase the discriminative power of the features. When compared to the K-SGF our solution always yields an increase in performance of almost $3\%$, which empirically validates the benefits of sampling along $\mathcal{C}^1$-smooth curves upon geodesic-based solutions. This increase is even larger when compared to the standard single-source domain approaches, performing on-par with the recent solution presented in [35], but using a much simpler approach. As expected, performance can be improved when more adequate solutions are used to obtain the domain's subspaces.

### B. Face recognition across facial expression and illumination variation

Face recognition experiments were conducted using images from two different datasets: 1) the Karolinska Directed Emotional Faces (KDEF) [37] and 2) the extended YaleB Face [38]. The Karolinska Directed Emotional Faces (KDEF) is a dataset of 4900 pictures of human facial expressions of

TABLE I

RECOGNITION ACCURACIES ON TARGET DOMAINS WITH *unsupervised* ADAPTATION (C: CALTECH, A: AMAZON, W: WEBCAM, AND D: DSLR). RESULTS OBTAINED CONSIDERING SUBSPACES WITH DIMENSION $k = 40$ AND A NUMBER OF $K = 12$ SAMPLED SUBSPACES PER GEOMETRIC SPLINE ($J = 25$). BEST RESULT IS SHOWN IN BRACKETS. *RESULTS REPORTED IN [35].

| Multi-Source | | | | Single Source | | | |
|---|---|---|---|---|---|---|---|
| $S^d \rightarrow$T | B-Geod+OLPP | $\mathcal{C}^1$-smooth+OLPP | K-SGF+OLPP | GFK-OLPP | CORAL* | ILS* | S→T |
| (A,C,W)→D | $89,5\%(89.9\%)$ | $90.8\%(91.7\%)$ | $86.3\%(88.4\%)$ | $93.3\%$ | $\mathbf{94.6}\%$ | $88.2\%$ | W→D |
| | | | | $66.4\%$ | $63.8\%$ | $67.1\%$ | C→D |
| | | | | $63.2\%$ | $61.3\%$ | $71.3\%$ | A→D |
| (D,C,W)→A | $84,6\%(85.9\%)$ | $85.8\%(86.2\%)$ | $83.4\%(83.8\%)$ | $80.9\%$ | $82.0\%$ | $86.7\%$ | W→A |
| | | | | $82.8\%$ | $\mathbf{88.6}\%$ | $87.1\%$ | C→A |
| | | | | $75.7\%$ | $71.2\%$ | $76.5\%$ | D→A |
| (A,C,D)→W | $93,1\%(94.0\%)$ | $93.2\%(93.8\%)$ | $91.2\%(91.4\%)$ | $\mathbf{94.1}\%$ | $93.5\%$ | $91.8\%$ | D→W |
| | | | | $80.5\%$ | $76.0\%$ | $80.1\%$ | C→W |
| | | | | $79.2\%$ | $71.8\%$ | $80.9\%$ | A→W |
| (A,D,W)→C | $78,5\%(79.1\%)$ | $\mathbf{80.8}\%(\mathbf{81.5}\%)$ | $73.9\%(74.1\%)$ | $73.4\%$ | $73.7\%$ | $76.3\%$ | W→C |
| | | | | $69.5\%$ | $63.0\%$ | $66.2\%$ | D→C |
| | | | | $71.9\%$ | $78.6\%$ | $78.4\%$ | A→C |

Fig. 3. Example images from the KDEF dataset (top) and extended YaleB dataset (bottom).

TABLE IV

RECOGNITION ACCURACIES ACROSS EXPRESSION VARIATION (ANGRY, DISGUSTED, AFFRAID, HAPPY, SAD, SURPRISED, NEUTRAL) USING GFK+PLS. RESULTS OBTAINED CONSIDERING SUBSPACES WITH DIMENSION $k = 100$. MEAN AVERAGE ACCURACY COMPUTE OVER ALL SOURCE↦TARGETS COMBINATIONS.

| GFK + PLS | Target Domains | | |
|---|---|---|---|
| Source Domain | Mean (Std) | Min | Max |
| AN | $64,2\%(\pm 7,4\%)$ | $52,5\%$ (SU) | $72,2\%$ (SA) |
| DI | $64,6\%(\pm 7,3\%)$ | $51,0\%$ (SU) | $71,5\%$ (SA) |
| AF | $69,4\%(\pm 3,2\%)$ | $64,5\%$ (AN) | $73,5\%$ (NE) |
| HA | $67,6\%(\pm 8,4\%)$ | $56,2\%$ (AN) | $81,6\%$ (NE) |
| SA | $63,4\%(\pm 8,7\%)$ | $51,3\%$ (SU) | $70,2\%$ (NE) |
| SU | $64,2\%(\pm 7,4\%)$ | $50,8\%$ (DI) | $73,8\%$ (NE) |
| NE | $71,0\%(\pm 5,6\%)$ | $62,8\%$ (DI) | $76,0\%$ (SA) |

emotions. The set contains images from 70 individuals of both gender, each displaying 7 different emotional expressions, each expression being photographed (twice) from 5 different angles. Frontal faces were used for evaluation, being cropped to $191 \times 186$ pixels and finally resized to $15\%$. The extended YaleB dataset contains 2414 frontal face images of 38 individuals of both gender. There are about 64 images for each person. The original images were cropped to $192 \times 168$ pixels and resized to $20\%$. This dataset is challenging due to varying illumination conditions.

We conducted two different experiments with these datasets: 1) Using the KDEF, we evaluate face recognition across facial expression variations, by clustering the faces in each one of the seven facial expressions (neutral, happy, sad, disgust, fear, hanger and surprise). For each expression's cluster, images from $50$ randomly selected individuals contributed to the training dataset while the images from the remaining 20 were used for testing. Each domain subspace frame was obtained using just the training datasets, which means that during training and smooth curve interpolation several individuals do not contribute with any sample to a certain domain. 2) Using the YaleB we evaluate face recognition across illumination variations, by clustering frontal faces acquired with illumination source located at one of the nine azimuth angles ($\pm 110°, \pm 70°, \pm 35°, \pm 20°, 0°$). Training was conducted using a reduced set of the evolving illumination clusters, being the remaining used for testing, which means that using interpolation intermediate illumination data can be synthesized.

In both experiments, each cluster represents a domain

TABLE V

RECOGNITION ACCURACIES ACROSS ILLUMINATION VARIATION (AZIMUTE ANGLES $\theta_i, i \in [0°, \pm 20°, \pm 35°, \pm 70°, \pm 110°]$) USING GFK+PLS. RESULTS OBTAINED CONSIDERING SUBSPACES WITH DIMENSION $k = 100$.

| GFK + PLS | Target Domains | | | | | |
|---|---|---|---|---|---|---|
| Source Domain | $-70°$ | $-35°$ | $-20°$ | $+20°$ | $+35°$ | $+70°$ |
| $\mathbf{-110°}$ | $71,9\%$ | $35.5\%$ | $23,4\%$ | $18,4\%$ | $14,4\%$ | $15,4\%$ |
| $\mathbf{0°}$ | $48,7\%$ | $83.9\%$ | $99,8\%$ | $99,6\%$ | $79,7\%$ | $46,4\%$ |
| $\mathbf{+110°}$ | $14,0\%$ | $16,7\%$ | $17,1\%$ | $21,4\%$ | $32,0\%$ | $62,5\%$ |

TABLE II

RECOGNITION ACCURACIES ACROSS EXPRESSION VARIATION (ANGRY, DISGUSTED, AFFRAID, HAPPY, SAD, SURPRISED, NEUTRAL). RESULTS OBTAINED CONSIDERING SUBSPACES WITH DIMENSION $k = 100$ AND A NUMBER OF $K = 12$ SAMPLED SUBSPACES FRAMES PER GEOMETRIC SPLINE.

| Source Domains | Target Domains | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **AN,DI,AF,HA,SA,SU** | AN | DI | AF | HA | SA | SU | NE | $\sum t_{d^i}$ |
| $\mathcal{C}^1$-smooth+PLS | $84,2\%$ | $86,7\%$ | $93,9\%$ | $92,3\%$ | $96,4\%$ | $83,0\%$ | $93,0\%$ | $89,0\%$ |
| K-SGF+PLS | $81,9\%$ | $84,4\%$ | $88,7\%$ | $88,1\%$ | $90,5\%$ | $78,9\%$ | $89,6\%$ | $86,7\%$ |

TABLE III

RECOGNITION ACCURACIES ACROSS ILLUMINATION VARIATION (AZIMUTE ANGLES $\theta_i, i \in [0°, \pm 20°, \pm 35°, \pm 70°, \pm 110°]$ ). RESULTS OBTAINED CONSIDERING SUBSPACES WITH DIMENSION $k = 100$ AND A NUMBER OF $K = 15$ SAMPLED SUBSPACES FRAMES PER GEOMETRIC SPLINE.

| Source Domains | Target Domains | | | | | | |
|---|---|---|---|---|---|---|---|
| $\mathbf{-110°,0°,+110°}$ | $-70°$ | $-35°$ | $-20°$ | $+20°$ | $+35°$ | $+70°$ | $\sum t_{d^i}$ |
| $\mathcal{C}^1$-smooth+PLS | $81,5\%$ | $85,6\%$ | $99,8\%$ | $97,5\%$ | $83,2\%$ | $81,2\%$ | $88,2\%$ |
| K-SGF+PLS | $72,8\%$ | $82,7\%$ | $98,7\%$ | $98,5\%$ | $78,8\%$ | $80,6\%$ | $83,9\%$ |

and several evaluations were conducted. We also synthesized additional domain shifts on frontal images by applying eight different levels of motion blur in the YALE dataset and six small in-plane face rotations in the KDEF dataset.

For the experiments conducted with the Yale database, the two most extreme illumination clusters ($\pm 110°$) and the frontal illumination cluster were used as source domains. In the KDEF experiments, all training sets were used as source domains and 10 different sets of training/test individuals were used to evaluate performance.

Results reported in Tables II-V confirm the benefits of sampling synthetic data along $\mathcal{C}^1$-smooth curves, when dealing with face recognition tasks in evolving events. In all experiments conducted, sampling along $\mathcal{C}^1$-smooth curves consistently yields top performance. When compared to single source domain adaptation solutions, our solution yields a significant increase in performance, above $5\%$ when compared with the K-SGF (multi-source domain), and an average increase above $20\%$ when compared with the Geodesic Flow Kernel approach (single-source domain). Note the good accuracy obtained with test conducted on NEutral face expression, a domain that was not included on the smooth interpolating curve.

### C. Stiefel Interpolation vs. Graßmann Interpolation

In contrast to the majority of solutions found in the literature, that mainly explore smooth interpolation in the Graßmann manifold, the algorithm proposed on this paper is the first one to explore smooth interpolation in the Stiefel manifold. In spite the mathematical complexity inherant to this manifold, performing interpolation in Stiefel has some advantages:

1) When compared to its dual solution in the Graßmann [23], $\mathcal{C}^1$-smooth interpolation in the Stiefel is computationally more efficient for low $k$-dimensional subspaces;
2) Depending on the metric used in the discriminative classifier, improved adaptation ability is observed when smooth interpolation is performed in Stiefel;
3) Complete smoothness of the interpolation curve is obtained in Stiefel, which is not guaranteed when the interpolation is performed in Graßmann [3].

Analyzing the computational costs involved in the interpolation algorithms in both manifolds, it is clear that for low $k$-dimensional subspaces computing $\mathcal{C}^1$-smooth curves in Stiefel is more efficient than in Graßmann. This efficiency vanishes as $k$ increases.

To evaluate the domain adaptation ability of the subspaces interpolation along both $\mathcal{C}^1$-smooth curves, face recognition across facial expression variation was conducted for the NEutral face expression domain, a domain that was not included in the smooth curve interpolation process. Two discriminative classifiers were used: KNN and PLS+KNN. Different metrics were considered for these two classifiers: Cosine, Euclidean and Standardized Euclidean; As can be observed in table VI, the adaptation ability when subspace interpolation is performed in Stiefel is beneficial depending on the classifier and also on the metrics used. The advantage of using smooth interpolating curves in Stiefel vanishes when inner-product based metrics are used in the KNN classifier. In the case of these metrics (Cosine and Euclidean) the projection map $\psi$ sends the $k$-frame $S_j$ into the subspace it generates, represented by $P_j = S_j S_j^{\mathrm{T}}$, which ends up being the same as using a smooth interpolating curve in Graßmann.

TABLE VI

DOMAIN ADAPTATION ABILITY USING INTERPOLATING CURVES IN THE STIEFEL AND GRASSMANN MANIFOLDS. RECOGNITION ACCURACIES ACROSS EXPRESSION VARIATION. SOURCE DOMAINS:{ AN,DI,AF,HA,SA,SU}. TARGET DOMAIN:{NE}

| | Classifiers & Metrics | | | | | |
|---|---|---|---|---|---|---|
| | cosine | | euclidean | | Stand. euclidean | |
| | KNN | PLS+KNN | KNN | PLS+KNN | KNN | PLS+KNN |
| $\mathscr{S}_{n,k}$ | $91,67\%$ | $93.3\%$ | $93,8\%$ | $94,3\%$ | $93,7\%$ | $95,3\%$ |
| $\mathscr{G}_{n,k}$ | $91,67\%$ | $92.7\%$ | $93,8\%$ | $94,0\%$ | $93,1\%$ | $92,4\%$ |

## VII. Conclusions

We have presented an alternative approach to solve the problem of multi-source domain adaptation via smooth interpolation on the Stiefel manifold. By sampling intermediate subspaces along smooth curves that interpolate the multiple domains, our solution is able to model more effectively the domain shifts between those domains. The generation of smooth interpolating curves in the Stiefel manifold was accomplished through the implementation of a recent intrinsic algorithm based on a modification of the Casteljau approach, where successive geodesic interpolation was replaced by successive quasi-geodesic interpolation.

The algorithm proposed in the paper may have a great impact in computer vision and related fields, since a curve that interpolates a set of points on the Stiefel manifold may correspond to the evolution of an event or dynamic scene from which only a limited number of observations was captured, envisaging new pathways for a multitude of related problems.

Experiments on cross-domain object recognition and face recognition across dynamically evolving datasets have been conducted in order to evaluate the proposed solution. On standard benchmark tasks, our solution consistently outperforms other competing algorithms, empirically validating the importance of our work.

## References


[1] Batzies, E., Hüper, K., Machado, L., Silva Leite, F.: Geometric mean and geodesic regression on Grassmannians. Linear Algebra and its Applications **466**(0) (2015) 83–101

[2] Edelman, A., Arias, T.A., Smith, S.T.: The Geometry of Algorithms with Orthogonality Constraints. SIAM J. Matrix Anal. Appl. **20**(2) (1998) 303–353

[3] Krakowski, K.A., Machado, L., Leite, F.S., Batista, J.: A modified casteljau algorithm to solve interpolation problems on stiefel manifolds. Journal of Computational and Applied Mathematics **311** (2017) 84 – 99

[4] Patel, V., Gopalan, R., Li, R., Chellappa, R.: Visual domain adaptation: a survey of recent advances. IEEE Signal Processing Magazine **32**(3) (2015) 53–69

[5] Gopalan, R., Li, R., Patel, V., Chellappa, R.: Domain adaptation for visual recognition. Foundations and Trends in Computer Graphics and Vision **8**(4) (2015) 285–378

[6] Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: IEEE Int. Conf. on Computer Vision (ICCV), Barcelona, SP (November 2011)

[7] Zheng, J., Liu, M., Chellapa, R., Phillips, P.: Grassmann manifold-based domain adaptation approach. In: APRP Int. Conf. on Pattern Recognition (ICPR), Tsukuba, JP (November 2012)

[8] Gong, B., Shi, Y. Sha, F., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Providence, RI (June 2012)

[9] Gong, B., Grauman, K., Sha, F.: Learning kernels for unsupervised domain adaptation with applications to visual object recognition. International Journal of Computer Vision **109**(1-2) (2014) 3–27

[10] Shrivastava, A., Shekhar, S., Patel, A.: Unsupervised domain adaptation using parallel transport on grasmann manifold. In: IEEE Winter Conf. Applications of Computer Vision, Steamboat Springs, CO (March 2014)

[11] Ni, J., Qui, Q., Chellappa, R.: Subspace interpolation via dictionary learning unsupervised domain adaptation. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, Portland, OR (2013)

[12] Hoffman, J., Darrell, T., Saenko, K.: Continuous manifold based adaptation for evolving visual domains. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR). (2014)

[13] Yang, Y., Hospedales, T.: A unified perspective on multi-domain and multi-task learning. In: International Conference on Learning Representations (ICLR 2015). (2015)

[14] Yang, Y., Hospedales, T.: Multivariate regression on the grassmannian for predicting novel domains. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016). (2016)

[15] Xu, X., H.T., Gong, S.: Multi-task zero-shot action recognition with prioritised data augmentation. In: European Conference on Computer Vision (ECCV 2016). (2016)

[16] Chopra, S., Balakrishnan, S., Gopalan, R.: Dlid: Deep learning for domain adaptation by interpolating between domains. In: International Conference on Machine Learning. (2013)

[17] Cui, Z., Li, W., Xu, D., Shan, S., Chen, X., Li, X.: Flowing on riemannian manifold: domain adaptation by shifting covariance. IEEE Transactions on Cybernetics **44**(12) (2014) 2264 – 2273

[18] Duan, L., Xu, D., Tsang, I.: Domain adaptation from multiple sources: a domain-dependent regularization approach. IEEE Trans. Neural Network Learning Systems **3**(23) (2014) 504–518

[19] Zhang, K., Gong, M., Schoelkopf, B.: Multi-source domain adaptation: A causal view. (2015)

[20] J., H., Kulis, B., Darrell, T., Saenko, K.: Discovering latent domains for multisource domain adaptation. In: European Conf. on Computer Vision (ECCV), Florence, IT (October 2012)

[21] Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation with multiple sources. In Koller, D., Schuurmans, D., Bengio, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 21. Curran Associates, Inc. (2009) 1041–1048

[22] Gopalan, R., Li, R., Chellappa, R.: Unsupervised adaptation across domain shifts by generating intermediate data representations. IEEE Trans. on Pattern Analysis and Machine Intelligence **36** (2014) 2288–2302

[23] Batista, J., Krakowski, K., Machado, L., Martins, P.: Multi-source domain adaptation using $c^1$-smooth subspaces interpolation. In: IEEE International Conference on Image Processing (ICIP). (Sep. 2016)

[24] Caseiro, R., Henriques, J., Martins, P., Batista, J.: Beyond the shortest path : Unsupervised domain adaptation by sampling subspaces along the spline flow. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), Boston, MA (June 2015)

[25] Su, J., Dryden, I.L., Klassen, E., Le, H., Srivastava, A.: Fitting smoothing splines to time-indexed, noisy points on nonlinear manifolds. Image Vision Comput. **30**(6-7) (June 2012) 428–442

[26] Horn, R.A., Johnson, C.R.: Topics in Matrix Analysis. Cambridge University Press, New York (1991)

[27] Gallivan, A., Srivastava, A., Liu, X., Van Dooren, P.: Efficient algorithms for inferences on grassmann manifolds. In: IEEE Workshop on Statistical Signal Processing. (Dec 2003) 315–318

[28] De Casteljau, P.: Outillages méthodes calcul. Technical Report (1959)

[29] Park, F., Ravani, B.: Bézier curves on Riemannian manifolds and Lie groups with kinematic applications. ASME Journal of Mechanical Design **117** (1995) 36–40

[30] Crouch, P., Kun, G., Silva Leite, F.: The de Casteljau algorithm on Lie groups and spheres. Journal of Dynamical and Control Systems **5**(3) (1999) 397–429

[31] Popiel, T., Noakes, L.: Bézier curves and $C^2$ interpolation in Riemannian manifolds. Journal of Aproximation Theory **148**(2) (October 2007) 111–127

[32] Nava-Yazdani, E., Polthier, K.: De Casteljau's algorithm on manifolds. Comput. Aided Geom. Des. **30**(7) (October 2013) 722–732

[33] Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: European Conf. on Computer Vision (ECCV), Crete, GR (September 2010)

[34] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: arXiv preprint arXiv:1409.1556v6. (2015)

[35] Herath, S., Harandi,M., Porikli, F.: Learning an invariant hilbert space for domain adaptation. In: arXiv preprint arXiv:1611.08350v1. (2016)

[36] Cai, D., He, X., Han, J., Zhang, H.J.: Orthogonal laplacianfaces for face recognition. IEEE Transactions on Image Processing **15**(11) (2006) 3608–3614

[37] Lundqvist, D., Flykt, A., Ohman, A.: The karolinska directed emotional faces - kdef. (1998)

[38] Georghiades, A., Belhumeur, P., Kriegman, D.: From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans. Pattern Anal. Mach. Intelligence **23**(6) (2001) 643–660