This CVPR2015 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Beyond the shortest path : Unsupervised Domain Adaptation by Sampling Subspaces along the Spline Flow

Rui Caseiro, João F. Henriques, Pedro Martins, and Jorge Batista

Institute of Systems and Robotics - University of Coimbra, Portugal
{ruicaseiro, henriques, pedromartins, batista}@isr.uc.pt

Abstract

Recently, a particular paradigm [18] in the domain adaptation field has received considerable attention by introducing novel and important insights to the problem. In this case, the source and target domains are represented in the form of subspaces, which are treated as points on the Grassmann manifold. The geodesic curve between them is sampled to obtain intermediate points. Then a classifier is learnt using the projections of the data onto these subspaces. Despite its relevance and popularity, this paradigm [18] contains some limitations. Firstly, in real-world applications, that simple curve (i.e. shortest path) does not provide the necessary flexibility to model the domain shift between the training and testing data sets. Secondly, by using the geodesic curve, we are restricted to only one source domain, which does not allow to take fully advantage of the multiple datasets that are available nowadays. It is then, natural to ask whether this popular concept could be extended to deal with more complex curves and to integrate multi-sources domains. This is a hard problem considering the Riemannian structure of the space, but we propose a mathematically well-founded idea that enables us to solve it. We exploit the geometric insight of rolling maps [30] to compute a spline curve on the Grassmann manifold. The benefits of the proposed idea are demonstrated through several empirical studies on standard datasets. This novel concept allows to explicitly integrate multi-source domains while the previous one [18] uses the mean of all sources. This enables to model better the domain shift and take fully advantage of the training datasets.

1. Introduction

In the past few years there has been a growing interest on the study and development of techniques, (e.g. domain adaptation, transfer learning [19, 34]) that enable the adaptation of classifiers to handle mismatches between the underlying distribution of the training and testing data (also known as source and target domains respectively).

Applications in computer vision often involve the study of real world problems where this scenario arises naturally (e.g. object/face/location recognition [3, 4, 26, 17, 13], object detection [25, 27], image segmentation [5, 6], discriminative face alignment [32]). The training and testing data can be acquired using sensors with different characteristics (e.g. image resolution or quality), under diverse lighting conditions (e.g. indoor controlled illumination vs outdoor environments) or from different camera viewpoints (e.g. same object in different poses).

For example, recently several important applications using domain adaptation were presented: Hoffman *et al.* [27] proposed a novel framework for large scale detection through adaptation (LSDA). They created a fast and effective large scale detection network by combining adaptation techniques with deep convolutional models. Fernando *et al.* [13] showed that using domain adaptation techniques is possible to deal with the image variability induced by large time lags. They presented a framework to automatically associate ancient pictures to modern ones, e.g. location recognition and interactive location retrieval. Ho *et al.* [26] demonstrated that using domain adaptation techniques is feasible to recognize faces in scenarios where images corresponding to the source and the target domain are acquired under varying degree of factors such as illumination, expression, blur and alignment. Finally, Gopalan *et al.* [17] presented a top-down approach through adaptation for estimating geographic location from images. They obtain competitive results for this challenging task compared with some landarmark papers in the field, e.g. [23].

This work was supported by the Portuguese Science Foundation (FCT) under the project Differential Geometry for Computer Vision Pattern Recognition (PTDC/EEA-CRO/122812/2010) and through the grants SFRH/BD74152/2010 (Rui Caseiro), SFRH/BD75459/2010 (João F. Henriques), SFRH/BPD/90200/2012 (Pedro Martins).

In this work, we explore the limitations of a popular and important paradigm in the field of domain adaptation [18, 19] and we propose some ideas to improve and extend it.

1.1. Related Work

Over the years, a vast number of paradigms for domain adaptation (with different variants) have been presented [36, 34, 18, 31, 16, 15, 28, 33, 12, 1, 14, 19]). Since a full-fledged description of all paradigms is beyond the scope of this paper, for the sake of brevity we will only outline some of the key ideas necessary to introduce the paradigm in study (please refer to [19] for a excellent survey).

Gopalan *et al*. [18, 19] proposed a novel concept for domain adaptation that has received noteworthy attention (SGF - Subspaces by Sampling Geodesic Flow). Drawing inspiration from incremental learning, instead of assuming information on the transformation or features across domains, it is considered that intermediate representations of data between the source and target domains provide a relevant description of the underlying domain shift. The global nature of these domains is represented in the form of generative subspaces, which are treated as points on the Grassmann manifold [40]. The geodesic curve between them is sampled to obtain intermediate points. Then a discriminative classifier is learnt using the projections of labeled data onto these subspaces. They deal with multiple sources by computing the mean on the Grassmann manifold (the mean is used as the source domain).

Gong *et al*. [16, 15] proposed the Geodesic Flow Kernel (GFK). The GFK models the domain shift by integrating along the geodesic an infinite number of intermediate subspaces. An infinite-dimensional feature vector is formed by projecting raw features into these subspaces. Inner products between these vectors give rise to a kernel function that can be computed over the original feature space in closed-form. To address the multiple source problem they proposed a metric called rank of domain that is used to choose one of them based on how suitable it is to domain adaptation.

Despite their relevance, SGF [18, 19] and GFK [16, 15] frameworks contain some limitations. Firstly, it is assumed that the best path to connect the source and target domains is the geodesic curve (shortest path). However, in real-world applications, this simple curve does not provide the necessary flexibility to model the domain shift between the training and testing datasets. Secondly, by using the geodesic curve, we are restricted to only one source domain, which does not allow us to fully take advantage of the multiple training data sets that are available nowadays in the community. Particularly in cases where the datasets contain several and different sources of variation, the subspace given by the mean point could not be the best option to represent the real characteristics of all the source domains. In fact in more challenge cases the path followed by the simple geodesic generate subspaces that do not allow us to take fully advantage of the training datasets.

One question arises: why does most research in this field focus on the geodesic curve?

We strongly believe that it is due the underlying Riemannian structure [37] of the Grassman manifold [11, 40, 7]. In general, the geodesic on a Riemannian manifold is a well defined and parameterized concept, while the definition of other more complex curves (e.g. splines) is a hard problem that has been in continuous study in the mathematics community. It is then natural to ask whether the popular SGF [18, 19] could be extended to deal with more complex curves and to integrate multi-sources.

1.2. Contributions

We propose a mathematically well-founded idea that allows to define a curve between more than two domains (see Fig. 7). As argued by Gopalan *et al*. [18, 19]: *humans adapt (better) between extreme domains if they gradually walk through the path between the domains* (e.g. [10]). Grounding in the same biological principle, we extend the SGF [18, 19] using smooth polynomial functions described by splines on the Grassmann manifold (SSF - Subspaces by Sampling Spline Flow).

We exploit the geometric insight of rolling maps [30, 37] to compute a spline curve on the Grassmann manifold. An association of unwrapping methods via local diffeomorphisms, and rolling motions (the manifold is rolled, without slipping or twisting, as a rigid body) enables us to map the data from the manifold to its affine tangent space at a point, compute the curve on the latter and then the final curve on the manifold is obtained by wrapping back while unrolling. The problem in the tangent space is solved using the De Casteljau's algorithm [35], which is a geometric method to construct smooth splines curves.

We remark that our goal is to improve/extend the SGF [18, 19] concept, so we used a sampling strategy along the spline to obtain intermediate subspaces. We believe that our idea could be propagated to other works, i.e. GFK [16, 15] (integrate all the points along the spline curve). However, to the best of our knowledge this is not possible in the moment and requires further research in differential geometry.

As we said before there exist a vast number of different paradigms and variants for domain adaptation. For example, using other paradigms several different variants have been proposed to deal with multi-sources (e.g. [28]). In order to mantain some focus in our work and for the sake of brevity the proof of concept is done evaluating directly against the SGF [18, 19], i.e. we do not compare with other multi-source techniques . We believe that is reasonable given the popularity of the SGF in the community.

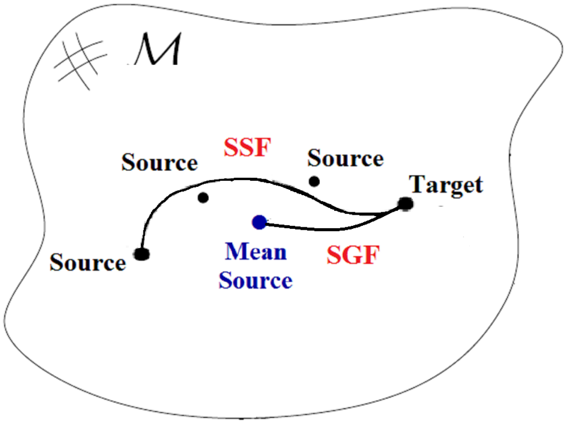

Figure 1. This image illustrates the SSF (Sampling Spline Flow) versus the SGF (Sampling Geodesic Flow) [18, 19]. The SGF deals with multiple sources by computing the respective mean and then a geodesic curve is defined between the mean-source and the target domain. The SSF uses smooth polynomial functions described by splines on the manifold to interpolate between all the sources and the target domain. The spline can be computed as an approximation (the curve passes close to the intermediate points) or exact spline [35]. The goal is to compute a curve able to extract more meaningful information from the sources.

2. Problem Description

In this section we outline the procedure to perform recognition under domain shift presented in [18, 19]. We adapted the formulation to include multi-sources.

We assume m source domains ($i = 0, .., m-1$) and one target domain ($i = m$). We denote the source data (labeled) as $X_i = \{x_{ij}\}_{j=1}^{N_i} \in \Re^D$ ($i = 0, .., m-1$).

Let $y_{ij} \in \{1, 2, 3, ...C\}$ be the label of x_{ij} (C classes). The data from the target (unlabeled) is denoted as $X_i = \{x_{ij}\}_{j=1}^{N_i} \in \Re^D$ ($i = m$), corresponding to the same C classes. The $D \times d$ subspaces S_i are obtained using principal component analysis on X_i ($i = 0, .., m$), ($d < D$) [18, 19].

The d-dimensional subspaces in $\Re^D$ will be treated as points in the Grassmann manifold $\mathcal{G}_{d,D}$.

The intermediate subspaces are obtained by sampling along the curve on the manifold, i.e. the geodesic curve as in SGF [18, 19] or a spline curve SSF (our work).

Let S' refer to a collection of m' subspaces including the m sources, the target and the intermediate subspaces. Let x'_{ij} represent the $dm' \times 1$ vector formed by concatenating the projection of x_{ij} onto all subspaces in S' [18, 19].

A discriminative classifier is trained using as input $D(X', Y')$, where X' is the $dm' \times N$ labeled data matrix (from sources) and Y' is the respective $N \times 1$ label's vector. The number of columns (N) of the matrix X' depend on the number of sources used to train the classifier, i.e. the curve and the respective set of subspaces S' are obtained using all the m source domains, but we can use as input to the classifier only the data from some source domains.

The procedure to build the classifier is the same that was used in SGF (please refer to [18, 19] for more technical details).

3. Rolling Maps on Riemannian Manifolds

The mathematical description of rolling motions of smooth manifolds has attracted some attention since the understanding on how to perform such virtual motions allows to estimate interpolating curves, solve optimal control and other complex problems on manifolds [30, 37, 38, 8]. Unlike other techniques for solving interpolation problems on manifolds, the association of rolling motions with unwrapping and wrapping technics, results in curves defined in explicit form.

Invoking the Whitney's theorem [21]: for a proper choice of $n \geq k$, a k-dimensional Riemannian manifold can be isometrically embedded into some Euclidean space $\Re^n$. Assuming the embedding space, rolling motions can be defined as rigid motions subject to specific constraints i.e. *holonomic* (rolling conditions) and *nonholonomic* (no-slip , no-twist conditions) [30, 37, 8].

Let $\mathcal{M}$ and $\bar{\mathcal{M}}$ be two oriented connected Riemannian manifolds with equal dimension both embedded in the same Euclidean space $\Re^n$ and let $\bar{\mathcal{M}}$ be the stationary manifold. The rolling motion describes how $\mathcal{M}$ rolls over $\bar{\mathcal{M}}$ without slip and twist [30, 37, 8].

The action of the special Euclidean group $\text{SE}_n = \text{SO}_n \ltimes \Re^n$ on $\Re^n$ is used to describe the rigid motion [30, 8]. The semi-direct product of the special orthogonal group $(\text{SO}_n, (\cdot))$ by the additive group $(\Re^n, (+))$ is represented by the symbol $\ltimes$.

It is considered that $\text{SO}_n \circ \mathbf{P} \subset \mathcal{M}$, for any $\mathbf{P} \in \mathcal{M}$, i.e. SO_n acts transitively on $\mathcal{M}$. Let $R \in \text{SO}_n$ be a rotation and $s \in \Re^n$ be a translation. Elements $h \in \text{SE}_n$ are usually defined by pairs $h = (R, s)$.

The action of SE_n on $\Re^n$ is represented as [30, 8] :

$$\text{SE}_n \times \Re^n \to \Re^n, \qquad (h, \mathbf{P}) \mapsto h \circ \mathbf{P} = R \circ \mathbf{P} + s. \quad (1)$$

Henceforth, the tangent space at $\mathbf{P} \in \mathcal{M}$ is denoted by $T_{\mathbf{P}}\mathcal{M}$ and $(T_{\mathbf{P}}\mathcal{M})^{\perp}$ represents the normal space (with respect to the Euclidean metric) [30, 8]. A rolling motion is characterized by a rolling map, which is a curve in SE_n that fulfills some requirements.

In this paper we consider the special case when $\bar{\mathcal{M}}$ is the affine tangent space $\mathcal{V} \cong T_{\mathbf{P}_0}\mathcal{M}$ (at the point $\mathbf{P}_0 \in \mathcal{M}$). The k-dimensional $\mathcal{V}$ is also a subspace embedded in $\Re^n$ [30, 8].

A *rolling map*, describing how $\mathcal{M}$ rolls upon $\bar{\mathcal{M}}$, without slip or twist, along a smooth *rolling curve* $\alpha\colon [0, T] \to \mathcal{M}$, is a smooth map (formal definition as given by Sharpe [37])

$$h\colon [0, T] \to \text{SE}_n = \text{SO}_n \ltimes \Re^n \qquad t \mapsto h(t) = (R(t), s(t)), \quad (2)$$

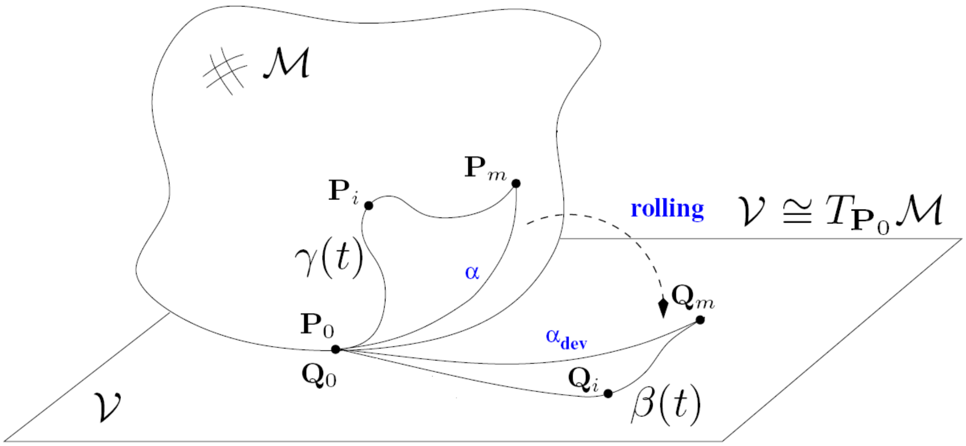

Figure 2. The manifold $\mathcal{M}$ rolls upon $\bar{\mathcal{M}} = \mathcal{V} \cong T_{\mathbf{P}_0}\mathcal{M}$ along a *rolling geodesic curve* $\alpha\colon [0,T] \to \mathcal{M}$. The case of the 2-dimensional sphere provides a nice physical intuition of the problem. When the sphere rolls over $\mathcal{V}$ it is forced to remain tangent to it by the holonomic constraints. The absence of sliding and spinning is the result of the nonholonomic conditions. $\beta(t)\colon [0,T] \to \mathcal{V}$ corresponds to the spline curve resulting of the De Casteljau's algorithm [35] on the tangent space (Euclidean space). $\gamma(t)\colon [0,T] \to \mathcal{M}$ corresponds to the spline curve on the manifold (obtained by wrapping back while unrolling the curve $\beta(t)$) [38, 8]. This image was adapted from the original presented in [38, 8].

satisfying the holonomic (rolling) and nonholonomic constraints (no-slip and no-twist).

The curve $\alpha_{\text{dev}}\colon [0,T] \to \bar{\mathcal{M}}$ given by $\alpha_{\text{dev}}(t) = h(t) \circ \alpha(t)$ is called the *development curve* of $\alpha(t)$ on $\bar{\mathcal{M}}$ (please refer to [30, 8] for more details).

4. Compute Spline Curves on a Riemannian Manifold

Let $\mathbf{S}_0, ..., \mathbf{S}_m \in \mathcal{M}$, be a set of $(m+1)$ domains where $\mathbf{S}_m$ represents the target domain and the others are the source domains. Recall that in the domain adaptation problem the domains are treated as points on the Grassmann manifold. The idea is to define a spline curve $\gamma(t)\colon [0,T] \to \mathcal{M}$ using those $(m+1)$ points.

Let T be a positive real number and $0 = t_0 < t_1 < ... < t_m = T$ a partition of the time interval $[0,T]$. The source domains $\mathbf{S}_0, ..., \mathbf{S}_{m-1} \in \mathcal{M}$ must be ordered such that $\gamma(t_i) = \mathbf{P}_i$ for the times t_i, where $0 = t_0 < t_1 < ... < t_{m-1}$ with $\gamma(t_0 = 0) = \mathbf{P}_0$, i.e. $\mathbf{P}_0, ..., \mathbf{P}_{m-1} \in \mathcal{M}$ are the ordered source domains. The target domain $\mathbf{S}_m$ is represented as $\gamma(t_m = T) = \mathbf{P}_m = \mathbf{S}_m$.

The problem is solved using an algorithm based on unwrapping methods via, local diffeomorphisms, and rolling motions (the manifold is rolled, without slipping or twisting, as a rigid body) [30, 8]. This combined technique projects the data from $\mathcal{M}$ onto $\mathcal{V} \cong T_{\mathbf{P}_0}\mathcal{M}$ (the goal is to project onto an Euclidean space) while mitigates the distortions normally associated with the simple projections done by using local diffeomorphisms only [30, 8].

In order to compute the curve we need to identify the affine tangent space $\mathcal{V}$ with $T_{\mathbf{P}_0}\mathcal{M}$ [30, 8]), then we define an orthonormal coordinate system for the tangent space and is computed a minimal (vectorial) representation $\mathbf{q}_0, ..., \mathbf{q}_m \in \Re^w$ of $\mathbf{Q}_0, ..., \mathbf{Q}_m$ [30, 8]). Since the $T_{\mathbf{P}_0}\mathcal{G}$ is an Euclidean space and is the space of symmetric matrices, there are only $w = D(D+1)/2$ independent coefficients. An orthonormal coordinate system for $T_{\mathbf{P}_0}\mathcal{G}$ is defined using this minimal representation [30, 8].

The vectorial representation $\mathbf{q}_0, ..., \mathbf{q}_m \in \Re^w$ is used to compute the spline curve on the tangent space (Euclidean space) $\beta(t)\colon [0,T] \to \mathcal{V}$ using the De Casteljau's algorithm (please refer to [35] for the implementation details) which is a geometric method to construct smooth splines curves. The spline can be computed as an approximation (the curve passes close to the intermediate points) or exact spline. Then the final curve on the manifold ($\gamma(t)\colon [0,T] \to \mathcal{M}$) is obtained by wrapping back while unrolling [30, 8].

Rolling/Unwrapping algorithm (refer to [30, 8]) :

1. Compute a curve $\alpha\colon [0,T] \to \mathcal{M}$ joining $\mathbf{P}_0$ $(t = 0)$ to $\mathbf{P}_m$ $(t = T)$ [30, 8].

2. Roll $\mathcal{M}$ over $\mathcal{V}$ along the curve $\alpha(t)$, using the rolling map $h(t)$. This generates a smooth curve $\alpha_{dev}\colon [0,T] \to \mathcal{V}$ joining the points $\mathbf{Q}_0 = \mathbf{P}_0$ and $\mathbf{Q}_m = h(t) \circ \mathbf{P}_m$ [30, 8].

3. Use a local diffeomorphism $\phi\colon \mathcal{M} \supset \Omega \to \mathcal{V}$, with $\mathbf{P}_0 \in \Omega$ [29], between $\mathcal{M}$ and the affine tangent space $\mathcal{V}$, satisfying $\phi(\mathbf{P}_0) = \mathbf{P}_0$, $D\phi(\mathbf{P}_0) = Id$, and project the data $\mathbf{P}_1, ..., \mathbf{P}_{m-1}$ to $\mathcal{V}$ while rolling along α. This combined technique of rolling and unwrapping projects the data $\mathbf{P}_1, ..., \mathbf{P}_{m-1}$ onto $\mathcal{V}$ to produce $\mathbf{Q}_1, ..., \mathbf{Q}_{m-1}$, given by [30, 8]

$$\mathbf{Q}_i = \phi((h(t_i) \circ \mathbf{P}_i) - \alpha_{\text{dev}}(t_i) + \mathbf{P}_0) + \alpha_{\text{dev}}(t_i) - \mathbf{P}_0. \quad (3)$$

4. Identify the affine tangent space $\mathcal{V}$ with $T_{\mathbf{P}_0}\mathcal{M}$, define an orthonormal coordinate system for the tangent space and compute a minimal (vectorial) representation $\mathbf{q}_0, ..., \mathbf{q}_m \in \Re^w$ of $\mathbf{Q}_0, ..., \mathbf{Q}_m$ [30, 8].

5. Compute the spline curve $\beta(t)\colon [0, T] \to \mathcal{V}$ using the data $\mathbf{q}_0, ..., \mathbf{q}_m$ and the De Casteljau's algorithm [35].

6. Wrap the curve $\beta(t)\colon [0, T] \to \mathcal{V}$ back onto the manifold using the formula :

$$\gamma(t) := h(t)^{-1}(\phi^{-1}(\beta(t) - \alpha_{\text{dev}}(t) + \mathbf{P}_0) + \alpha_{\text{dev}}(t) - \mathbf{P}_0). \tag{4}$$

5. Rolling Maps for Grassmann Manifolds

In this section we present the rolling maps for the Grassmann [30, 8]. We focus on the kinematic equations for rolling the Grassmann on the affine tangent space at a point.

These equations were derived using the general definition of rolling and the respective holonomic and nonholonomic constraints. Remarkably, they are particularly simple since Hupper *et al.* [30, 8] managed to adapt the rolling in order to preserve the matrix struture of the Grassmann.

The Grassmann manifold of all d-dimensional subspaces of $\Re^D$ is denoted as $\mathcal{G}_{d,D}$ (simplified notation = $\mathcal{G}$). In general this manifold is represent using a $D \times d$ matrix, however in order to define the Rolling maps for the Grassmann, Hupper *et al.* [30], used a special representation for this particular manifold.

The Grassmann manifold is defined as a particular subset of the symmetric matrices Sym_D, i.e. a d-dimensional subspace in $\Re^D$ is uniquely associated with an $(D \times D)$ orthogonal projection matrix $H = H^T$ of rank d [30, 8, 24],

$$\mathcal{G}_{d,D} := \{H \in \text{Sym}_D \mid H^2 = H, \ \text{rank}(H) = d\}. \tag{5}$$

Note : in Section 6 we will explain more details about this operator [24].

The space of Sym_D, equipped with the metric induced by the Frobenius norm (Euclidean metric for matrices) is used to embed $\mathcal{G}_{d,D}$ and the affine tangent space at a point.

The *rolling motion* is described by the action of the group $G = \text{SO}_D \ltimes \text{Sym}_D$ on $\mathbf{S} \in \text{Sym}_D$ by the rule [30, 8]

$$G \times \text{Sym}_D \to \text{Sym}_D, \qquad ((\Theta, X), \mathbf{S}) \mapsto \Theta\mathbf{S}\Theta^T + X, \tag{6}$$

where $\Theta \in \text{SO}_D$ and $X \in \text{Sym}_D$.

Given $\Theta(t) \in \text{SO}_D$ satisfying $\Theta(0) = I$ (identity matrix) and $\alpha(0) = \mathbf{P}_0$, a smooth curve $\alpha\colon [0, T]$ in $\mathcal{G}$ is defined as [30, 8]

$$t \to \alpha(t) = \Theta(t)\mathbf{P}_0\Theta^T(t). \tag{7}$$

In this moment the objective is to determine conditions on the mapping [30, 8]

$$h\colon [0, T] \to \text{SO}_D \ltimes \text{Sym}_D, \quad t \to h(t) = (\Theta^T(t), X(t)), \tag{8}$$

so that it is a rolling map of $\mathcal{G}$ over the affine tangent space at $\mathbf{P}_0 \in \mathcal{G}$, along the curve $\alpha(t)$, and development curve $\alpha_{\text{dev}}(t) = h(t) \circ \alpha(t)$ is given by [30, 8]

$$\alpha_{\text{dev}}(t) = \Theta^T(t)\alpha(t)\Theta(t) + X(t) = \mathbf{P}_0 + X(t). \tag{9}$$

Note that $\Theta\mathbf{S}\Theta^T \in \mathcal{G}$ for any $\mathbf{S} \in \mathcal{G}$.

Assuming the base point $\mathbf{P}_0$ equal to

$$\mathbf{P}_0 = \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix}, \tag{10}$$

, the kinematic equations are as follows (where $\Theta(0) = I$, $X(0) = 0$) [30, 8]

$$\dot{X}(t) = \begin{bmatrix} 0 & \Psi(t) \\ \Psi^T(t) & 0 \end{bmatrix}, \ \dot{\Theta}(t) = \Theta(t)\begin{bmatrix} 0 & -\Psi(t) \\ \Psi^T(t) & 0 \end{bmatrix}. \tag{11}$$

A control function can be defined using the (matrix) function $\Psi\colon \Re \to \Re^{d\times(D-d)}$. The rolling curve can be chosen using Ψ. Assuming $\Psi(t) = \Psi$ constant the kinematic equations can be computed explicitly as [30, 8]

$$\Theta(t) = \exp\left(t\begin{bmatrix} 0 & -\Psi \\ \Psi^T & 0 \end{bmatrix}\right), \ X(t) = t\begin{bmatrix} 0 & \Psi \\ \Psi^T & 0 \end{bmatrix}, \tag{12}$$

and, in this situation, the rolling α and the development α_{dev} curves are geodesics in $\mathcal{G}$ and in the affine tangent space, respectively [30, 8].

The exponential of matrices with a special block structure as in eq. (12) can be computed explicitly [30, 8]. Using this fact, Θ can be given by (where B is represented by a series expansion) [30, 8]

$$\Theta(t) = \begin{bmatrix} (I_k - BB^T)^{1/2} & -B \\ B^T & (I_{n-k} - B^TB)^{1/2}, \end{bmatrix}, \tag{13}$$

with $B := \Psi\dfrac{\sin(\Psi^T\Psi)^{1/2}}{(\Psi^T\Psi)^{1/2}}$.

Given the initial $\alpha(0) = \mathbf{P}_0$ and final $\alpha(T) = \mathbf{P}_m$ points of the geodesic curve defined by $\alpha(T) = \mathbf{P}_m = \Theta(T)\mathbf{P}_0\Theta^T(T)$ we need to calculte the matrix Ψ [30, 8].

This is not a simple problem, since generally the matrices $\mathbf{P}_0$ and $\mathbf{P}_m$ are not invertible [30, 8]. However, Srivastava *et al.* [39] proposed an elegant solution to this task (refer to [39] for more details).

All the mathematics involving rolling can be easily recalculated if the base point is not the particular point $\mathbf{P}_0$ used above [30]. Indeed, since any other point $\mathbf{P} \in \mathcal{G}$ is related to the point $\mathbf{P}_0$ by $\mathbf{P} = Q\mathbf{P}_0Q^T$, for some $Q \in \text{SO}_D$, it also happens that if $(X(t), \Theta(t))$ is the solution of the kinematic equations (above) for the base point $\mathbf{P}_0$, then $(QX(t)Q^T, Q\Theta(t)Q^T)$ is the solution of the kinematic equations for rolling the Grassmann manifold when the base point is $\mathbf{P} = Q\mathbf{P}_0Q^T$ [30].

6. Projection Operator and its Inverse

As we have said before, in general the Grassmann manifold $\mathcal{G}_{d,D}$ is represented using a $D \times d$ matrix, which corresponds to the subspaces computed using PCA. However in order to simplify the Rolling maps for the Grassmann, Hupper *et al*. [30] used a different representation.

The Grassmann manifold is defined as a particular subset of the symmetric matrices Sym_D, i.e. a point in $\mathcal{G}_{d,D}$ is uniquely associated with an $(D \times D)$ orthogonal projection matrix $H = H^T$ of rank d (idempotent symmetric matrices with rank d) $\{H \in \text{Sym}_n \mid H^2 = H, \ \text{rank}(H) = d\}$ [30, 24]. This projection embedding has been previously applied in other applications, such as meanshift [9], sparse coding [22] or subspace tracking [39].

Two questions arise : how to define this projection operator and how to map back to $\mathcal{G}_{d,D}$.

After the sampling along the spline curve on the manifold we need to map back the points from the space $\{H \in \text{Sym}_n \mid H^2 = H, \ \text{rank}(H) = d\}$ to $\mathcal{G}_{d,D}$ in order to apply the subspaces to the data. To this end it is necessary to define the respective function as a diffeomorphism i.e. a one-to-one, continuous, differentiable mapping with a continuous, differentiable inverse

Following the Helmke's book [24] : if $\mathcal{H}_d(D)$ denotes the set of all Hermitian projection operators $\mathcal{H}$ of $\Re^D$ with rank d ([24] p. 9)

$$H = H^T, \qquad H^2 = H, \qquad \text{rank}(H) = d \tag{14}$$

then, by the spectral theorem, every $\mathcal{H} \in \mathcal{H}_d(D)$ is of the form $H = X \cdot X^T$ for a real $D \times d$-matrix X satisfying $X^T X = I_d$.

The map $f : \mathcal{H}_d(D) \to \mathcal{G}(d,D)$ ([24] p. 9) is a bijection. Let $\Lambda = \text{diag}(\lambda_1, ..., \lambda_D)$ with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D$ be a real diagonal matrix and let ([24] p. 44)

$$M(\Lambda) = \{\Theta' \Lambda \Theta \in \Re^{D \times D} | \Theta\Theta' = I_D\}, \tag{15}$$

denote the set of all real symmetric matrices $H = \Theta' \Lambda \Theta$ orthogonally equivalent to Λ. Thus $M(\Lambda)$ is the set of all symmetric matrices with eigenvalues $\lambda_1, ..., \lambda_D$. For

$$\Lambda = \begin{bmatrix} I_d & 0 \\ 0 & 0 \end{bmatrix}, \tag{16}$$

$M(\Lambda)$ coincides with the set of all rank d symmetric projection operators H of $\Re^D$ ([24] p. 63).

Following the Lemma 2.5 in ([24] p. 63), with this condition the isospectral manifold $M(\Lambda)$ is diffeomorphic to the Grassmann $\mathcal{G}_{d,D}$. To any orthogonal $D \times D$ matrix ([24] p. 63)

$$\Theta = \begin{bmatrix} \Theta_1 \\ \Theta_2 \end{bmatrix} \qquad \Theta_1 \in \Re^{d \times D} \qquad \Theta_2 \in \Re^{(D-d) \times D} \tag{17}$$

we associate the d dimensional vector-space $V_{\Theta_1} \subset \Re^D$, which is generated by the d orthogonal row vectors of Θ_1.

This defines a map ([24] p. 63)

$$f : M(\Lambda) \ \to \ \mathcal{G}(d,D) \qquad \Theta' \Lambda \Theta \ \mapsto \ V_{\Theta_1} \tag{18}$$

f is a bijection and a diffeomorphism ([24] p. 63). In fact, as $M(\Lambda)$ is the set of rank d symmetric projection operators, $f(H) \in \mathcal{G}_{d,D}$ is the image of $H \in M(\Lambda)$ ([24] p. 64).

Let $X \in \Re^{D \times d}$ be such that the columns of X generate a d-dimensional linear subspace $V \subset \Re^D$. Then $H = X(X'X)^{-1}X'$ is the Hermitian projection operator ([24] p. 64).

7. Experimental Results

The proposed method (**SSF**) was evaluated for unsupervised domain adaptation in the context of visual object recognition [18, 19, 16, 36].

We compare our approach with the **SGF** [18] in which the intermediate subspaces are sampled from a geodesic curve and the multi-source domain problem is solved using the mean of all source domains. In order to mantain some focus in our work and for the sake of brevity the proof of concept is done evaluating directly against the SGF, i.e. we do not compare with other multi-source techniques. The main goal is to demonstrate that by considering more complex curves and with the integration of multi-source domains we obtain a paradigm more flexible that provides better results. This comparison, serve principally as a proof of concept, which we hope reasonable given the novelty of the technique.

We divided the experiments in two modes:

Mode 1: **SSF** - the spline is defined using all sources / the classifier is trained using the data of all sources ; **SGF** - the source used to define the geodesic is the mean of all sources / the classifier is trained using the data of all sources.

Mode 2: **SSF** - the spline is defined using all sources / the classifier is trained using only the data available in the **Source 1** ; **SGF** - the source used to define the geodesic is the **Source 1** / the classifier is trained using only the data available in the **Source 1**.

Across all the experiments we used 10-15 intermediate points (in each experiment was used a equal number of intermediate points for **SGF** and **SSF** cases).

We used in our tests on the task of visual object recognition the dataset of [36] that has 31 different object categories collected under three domain settings (4652 images in total): *Amazon* (images downloaded from online merchants), *DSLR* (high-resolution images by a digitalSLR camera), and *Webcam* (low-resolution images by a web camera). The Amazon dataset contains images acquired in highly-controlled conditions (studio illumination). The high resolution images of the DSLR dataset were acquired in a home

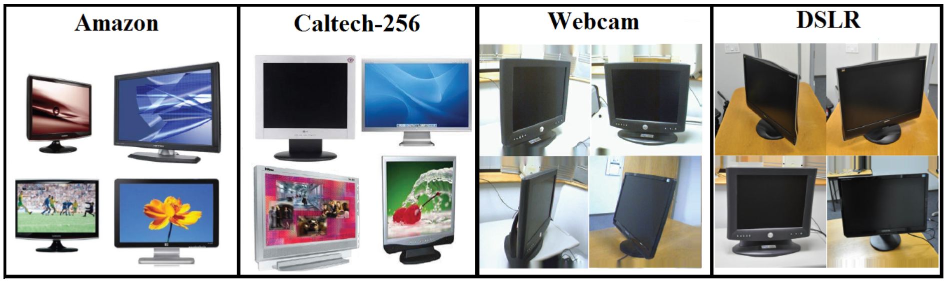

Figure 3. Images examples from the four datasets (Amazon, Caltech-256, Webcam, DSLR) used (Monitor category).

	SSF - **Spline** (Our)					**SGF** - **Geodesic** [18]	
Target ($\mathbf{P}_3$)	**Source 1** ($\mathbf{P}_0$)	**Source 2** ($\mathbf{P}_1$)	**Source 3** ($\mathbf{P}_2$)	**Mode 1**	**Mode 2**	**Mode 1**	**Mode 2**
						(Source = Mean)	(Source = Source 1)
Amazon (A)	DSLR	Webcam	Caltech	48.2	40.9	Mean (C , D , W)	31.2
Amazon (A)	Webcam	DSLR	Caltech	46.1	37.0	34.5	28.4
Caltech (C)	DSLR	Webcam	Amazon	44.9	38.5	Mean (A , D , W)	28.4
Caltech (C)	Webcam	DSLR	Amazon	43.1	31.9	31.9	22.8
DSLR (D)	Amazon	Caltech	Webcam	62.1	42.8	Mean (A , C , W)	31.0
DSLR (D)	Caltech	Amazon	Webcam	60.8	43.1	44.1	32.8
Webcam (W)	Caltech	Amazon	DSLR	65.8	41.5	Mean (A , C , D)	30.7
Webcam (W)	Amazon	Caltech	DSLR	65.1	40.9	45.2	32.0

Table 1. Classification accuracies on visual object recognition (**Data 1**).

environment (natural lighting) using a digital SLR camera. The lower resolution images of the Webcam dataset were taken in a analogous ambience as the DSLR ones, but include color and white balance artifacts as well substancial noise. The DSLR and Webcam datasets have roughly around 30 instances for a category whereas the Amazon has a average of 90 instances for each category.

The well-known *Caltech-256* [20] was also included as a fourth dataset. It contains images of 256 object classes.

We followed a standard protocol for extracting and coding image features to represent the objects and for the train/test procedure [18, 19, 16, 36]. All images were resized to the same dimensions and converted to grayscale. We extracted 64-dimensional SURF features [2], with a fixed blob response threshold. The images were encoded using a codebook of size 800 generated by k-means clustering on a subset of Amazon database. Then for all domains the images were encoded by a 800 bin histogram. The histograms were normalized and then z-scored in order to each dimension have zero mean and unit standard deviation. From this data ($D = 800$) the subspaces were computed using PCA. Across the experiments the value of d was set to 175-225 and we used 10-15 intermediate points (in each experiment was used a equal value d and number of intermediate points for **SGF** and **SSF** cases). The accuracy is reported over 20 different trials.

We divided the experiments on the visual object recognition task in two parts caracterized by different settings of data used : **Data 1**: we use all the four datasets (Amazon, Caltech256, DSLR, Webcam) and we extracted 10 classes common to them : Backpack, Touring-Bike, Computer-Keyboard, Laptop-101, Computer-Monitor, Computer-Mouse, Calculator, Head-Phones, Coffee-Mug and Video-Projector. There are in total 2533 images (8 to 151 samples per category per dataset). **Data 2**: we use three datasets (Amazon, DSLR, Webcam) and we extracted all the 31 classes The results are reported in Tables 1 and 2.

An important step in our algorithm is to define an order for the sources ($\mathbf{P}_0, ..., \mathbf{P}_{m-1}$). To address this task we used a metric called rank of domain proposed in [16]. This metric can be used to order the source domains based on how suitable they are to domain adaptation. For example, when the target is the *Amazon* ($\mathbf{P}_3$) the best source is the *Caltech* ($\mathbf{P}_2$) followed by the *Webcam* ($\mathbf{P}_1$) and by the *DSLR* ($\mathbf{P}_0$) (see Table 4 [16]).

It can be seen that in **Mode 1** the results empirically demonstrate the claims of our work. The benefits of the proposed **SSF** over the **SGF** are significative. Even in **Mode 2**, our paradigm outerperforms the **SGF**, since the data contained in the **Source 1** is better adapted using the subspaces sampled from the spline. The spline allows to explicitly integrate the multi-source domains while the **SGF** uses the mean of all the sources. This enables to model better the domain shift and take fully advantage of all the trainning datasets.

8. Conclusions and Future Work

To the best of our knowledge, this is the first work that proposes an avenue that allows to extended the popular paradigm proposed by Gopalan *et al.* [18, 19] (SGF) to deal

	SSF - Spline Curve (Our)				**SGF - Geodesic Curve** [18]	
Target ($\mathbf{P}_2$)	**Source 1** ($\mathbf{P}_0$)	**Source 2** ($\mathbf{P}_1$)	**Mode 1**	**Mode 2**	**Mode 1**	**Mode 2**
					(Source = Mean)	(Source = Source 1)
Amazon (A)	DSLR	Webcam	40.2	36.1	Mean (D , W)	29.0
Amazon (A)	Webcam	DSLR	37.5	31.4	31.5	27.5
DSLR (D)	Amazon	Webcam	49.6	29.5	Mean (A , W)	22.4
DSLR (D)	Webcam	Amazon	37.3	45.9	33.9	40.2
Webcam (W)	Amazon	DSLR	47.8	38.1	Mean (A , D)	32.8
Webcam (W)	DSLR	Amazon	44.6	43.3	37.5	38.6

Table 2. Classification accuracies on visual object recognition (**Data 2**).

with polynomial curves on the Grassman manifold and to integrate multi-source domains. We proposed a mathematically well-founded idea to do it by exploiting the geometric insight of rolling maps [30, 37] to describe smooth polynomial functions using splines (SSF). Our paradigm provides more flexibility to model the domain shift between the train/test datasets, and allow us to fully take advantage of the multiple training datasets that are available nowdays.

We believe that the proposed SSF could be useful to solve several other problems in vision. Recently, the Gopalan's work (SGF) [18, 19] has been also used with sucess for estimating geographic location from images [17, 23] and for unconstrained face recognition [26]. The SSF can be a promising solution to improve those challenging problems.

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, 2013. 2

[2] H. Bay, T. Tuytelaars, and L. Gool. Surf: Speedup robust features. In *ECCV*, 2006. 7

[3] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*, 2012. 1

[4] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Free-form region description with second-order pooling. In *PAMI*, 2014. 1

[5] R. Caseiro, J. F. Henriques, and J. Batista. Using directional statistics to learn cast shadows from a multi-spectral light sources physical model. In *ICIP*, 2010. 1

[6] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista. A non-parametric riemannian framework on tensor field application to foreground segmentation. In *ICCV*, 2011. 1

[7] R. Caseiro, J. F. Henriques, P. Martins, and J. Batista. Semi-intrinsic mean shift on riemannian manifolds. In *ECCV*, 2012. 2

[8] R. Caseiro, P. Martins, J. F. Henriques, F. S. Leite, and J. Batista. Rolling riemannian manifolds to solve the multi-class classification problem. In *CVPR*, 2013. 3, 4, 5

[9] H. Cetingul and R. Vidal. Intrinsic mean shift for clustering on stiefel and grassmann manifolds. In *CVPR*, 2009. 6

[10] S. Chalup. Incremental learning in biological machine learning systems. *Int. Journal Neural Systems*, 12:447–466, 2002. 2

[11] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Analysis Applications*, 20(2):303–353, 1998. 2

[12] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *CVPR*, 2013. 2

[13] B. Fernando, T. Tommasi, and T. Tuytelaars. Lost in the past: Recognizing locations over large time lags. In *arxiv*, 2014. 1

[14] B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *NIPS*, 2013. 2

[15] B. Gong, K. Grauman, and F. Sha. Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *IJCV*, 109(1-2):3–27, 2014. 2

[16] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012. 2, 6, 7

[17] R. Gopalan. Learning cross-domain information transfer for location recognition and clustering. In *CVPR*, 2013. 1, 8

[18] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011. 1, 2, 3, 6, 7, 8

[19] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *PAMI*, 2014. 1, 2, 3, 6, 7, 8

[20] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. In *Technical report*, 2007. 7

[21] V. Guillemin and A. Pollack. *Differential Topology*. Prentice-Hall, 1974. 3

[22] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovel. Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution. In *ICCV*, 2013. 6

[23] J. Hays and A. Efros. Im2gps: estimating geographic information from a single image. In *CVPR*, 2008. 1, 8

[24] U. Helmke and J. B. Moore. *Optimization and Dynamical Systems*. 1996. 5, 6

[25] J. Henriques, J. Carreira, R. Caseiro, and J. Batista. Beyond hard negative mining: efficient detector learning via block-circulant decomposition. In *ICCV*, 2013. 1

[26] H. Ho and R. Gopalan. Model-driven domain adaptation on product manifolds for unconstrained face recognition. 109(110–125):IJCV, 2014. 1, 8

[27] J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *NIPS*, 2014. 1

[28] J. Hoffman, B. Kulis, T. Darrell, , and K. Saenko. Discovering latent domains for multisource domain adaptation. In *CVPR*, 2012. 2

[29] K. Huper, M. Kleinsteuberb, and H. Shenb. Averaging complex subspaces via a karcher mean approach. *Signal Processing*, 93(459–467), 2013. 4

[30] K. Hupper and F. S. Leite. On the geometry of rolling and interpolation curves on Sn, SOn, Grassmann manifolds. *J. Dynamical Control Systems*, 13(4):467–502, 2007. 1, 2, 3, 4, 5, 6, 8

[31] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 2

[32] P. Martins, R. Caseiro, J. Henriques, and J. Batista. Discriminative bayesian active shape models. In *ECCV*, 2012. 1

[33] J. Ni, Q. Qiu, and R. Chellappa. Subspace interpolation via dictionary learning unsupervised domain adaptation. In *CVPR*, 2013. 2

[34] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 1, 2

[35] R. Rodrigues and F. Leite. A new geometric algorithm to generate smooth interpolating curves on riemannian manifolds. *LMS J. Comput. Math.*, 8:251–266, 2005. 2, 3, 4, 5

[36] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2, 6, 7

[37] R. Sharpe. *Differential Geometry*. Springer, 1996. 2, 3, 8

[38] Y. Shen, K. Hupper, and F. S. Leite. Smooth interpolation of orientation by rolling and wrapping for robot motion planning. In *ICRA*, 2006. 3, 4

[39] A. Srivastava and E. Klassen. Bayesian and geometric subspace tracking. *Adv. in Applied Probability*, 36(1):43–56, 2004. 5, 6

[40] P. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa. Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *PAMI*, 33(11):2273–2286, 2011. 2