Int J Comput Vis (2017) 122:388–408
DOI 10.1007/s11263-016-0941-2

3D Human Pose Tracking Priors using Geodesic Mixture Models

Edgar Simo-Serra[1] · **Carme Torras**[2] · **Francesc Moreno-Noguer**[2]

Received: 10 August 2015 / Accepted: 12 August 2016 / Published online: 24 August 2016

Abstract We present a novel approach for learning a finite mixture model on a Riemannian manifold in which Euclidean metrics are not applicable and one needs to resort to geodesic distances consistent with the manifold geometry. For this purpose, we draw inspiration on a variant of the expectation-maximization algorithm, that uses a minimum message length criterion to automatically estimate the optimal number of components from multivariate data lying on an Euclidean space. In order to use this approach on Riemannian manifolds, we propose a formulation in which each component is defined on a different tangent space, thus avoiding the problems associated with the loss of accuracy produced when linearizing the manifold with a single tangent space. Our approach can be applied to any type of manifold for which it is possible to estimate its tangent space. Additionally, we consider using shrinkage covariance estimation to improve the robustness of the method, especially when dealing with very sparsely distributed samples. We evaluate the approach on a number of situations, going from data clustering on manifolds to combining pose and kinematics of articulated bodies for 3D human pose tracking. In all cases, we demonstrate remarkable improvement compared to several chosen baselines.

Keywords Probabilistic priors · Mixture modelling · Riemannian manifolds · 3D human pose · Human kinematics

Communicated by Hiroshi Ishikawa, Takeshi Masuda, Yasuyo Kita and Katsushi Ikeuchi.

✉ Edgar Simo-Serra
esimo@aoni.waseda.jp

Carme Torras
torras@iri.upc.edu

Francesc Moreno-Noguer
fmoreno@iri.upc.edu

[1] Waseda University, Okubo 3-4-1, Shinjuku, Tokyo 169-8555, Japan

[2] Institut de Robòtica i Informàtica Industrial (CSIC-UPC), 08028 Barcelona, Spain

1 Introduction

The use of Riemannian manifolds and their statistics has recently gained popularity in a wide range of applications involving non-linear data modeling. For instance, they have been used to model shape changes in the brain Davis et al. (2007), diffusion tensor imaging Pennec et al. (2006), deformations of anatomical parts Fletcher et al. (2004) and human motion Brubaker et al. (2012), Sommer et al. (2010). In this work we tackle the problem of approximating the probability density function (PDF) of a potentially large dataset that lies on a *known* Riemannian manifold. We address this by creating a completely data-driven algorithm consistent with the manifold, i.e., an algorithm that yields a PDF defined on the manifold. This PDF can then be used as a prior in higher-order models, by combining it with image evidence in hybrid discriminative-generative models Simo-Serra et al. (2013), or by exploiting it to constrain the search space in a tracking framework Andriluka et al. (2010). We will show particular applications of the proposed prior in the case of 3D human pose estimation, demonstrating a remarkable improvement compared to other widely used models.

A standard procedure to operate on a manifold is to use the logarithmic map to project the data points onto the tangent space of the mean point on the manifold Fletcher et al. (2004), Huckemann et al. (2010), Sommer et al. (2010). After this linearization, Euclidean statistics are computed and projected back to the manifold using the exponential map. This

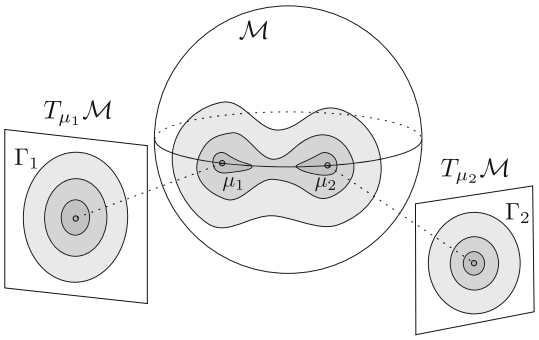


Fig. 1 Illustration of the proposed mixture model approach. Each mixture component has its own tangent space, ensuring the consistency of the model while minimizing accuracy loss

process is iteratively repeated until convergence of the computed statistics. Unfortunately, while this approximation is effective to model data with a reduced extent, it is prone to fail when dealing with data that covers wide regions of the manifold.

In the proposed finite mixture model, we overcome this limitation by simultaneously considering multiple tangent spaces, distributed along the whole manifold as seen in Fig. 1. We draw inspiration on the unsupervised algorithm from Figueiredo and Jain (2002), which given data lying in an Euclidean space, automatically computes the number of model components that minimizes a message length cost. By representing each component as a distribution on the tangent space at its corresponding mean on the manifold, we are then able to generalize the algorithm to Riemannian manifolds and at the same time mitigate the accuracy loss produced when using a single tangent space. Furthermore, since our model is *semi-parametric*, we can handle an arbitrarily large number of samples. This is in contrast to existing *non-parametric* approaches Pelletier (2005) whose complexity grows with the training set size.

As an example of practical application of our mixture model, we will consider the 3D human pose tracking problem, which has been traditionally addressed with kinematic priors based on Gaussian diffusion Deutscher and Reid (2005), Gall et al. (2010), Hauberg et al. (2012), Sigal et al. (2012), Sminchisescu and Triggs (2003). This consists in simply searching in a small area defined by a Gaussian distribution centered on the previous pose, i.e., $x_t = x_{t-1} + \epsilon$, where x_t would be the pose at time t and ϵ would be a Gaussian perturbation with 0 mean and diagonal covariance. However, this simple model does not constrain the pose to lie on its underlying manifold, and does indeed explore a much higher dimensional space than it should be strictly necessary. We will show that using our model as a kinematic prior we can effectively focus our solution on the actual manifold, greatly outperforming standard Gaussian diffusion models.

A preliminary version of this work appeared in Simo-Serra et al. (2014) with an application to introducing kinematic priors later presented in Simo-Serra et al. (2015). We extend this work by considering improvements for the covariance estimation. In particular, we consider shrinkage estimators that are shown to outperform empirical covariance estimation when the samples are sparsely distributed on the manifold (because the manifold has a very large dimensionality or because the number of samples is small, or a combination of both effects). This makes our approach both appropriate to handle situations with either large or small amounts of data, while our previous versions were mostly effective when dealing with large datasets. We finally unify and extend the evaluation in Simo-Serra et al. (2014; 2015) to consider more manifolds and the improvements proposed in this paper. Results will show that our manifold-based finite mixture model can be used to exploit the known structure of the data, outperforming approaches that do not. We provide the source code[1] of our approach.

2 Related Work

Manifolds have always been very important in computer vision Sanin et al. (2012). The two more widely used manifolds have been the one of symmetric semi-definite matrices Harandi et al. (2012; 2014), Jayasumana et al. (2013; 2015), Pennec (2009), Sivalingam et al. (2010), and the Grassman manifolds Jain and Govindu (2013), Shirazi et al. (2012), Turaga et al. (2011). However, most of these approaches focus on exploiting very specific manifolds and do not generalize to other manifolds. In contrast, our approach is applicable to all Riemannian manifolds with explicit exponential and logarithmic maps.

Recently, there has been an influx of theoretical results in statistics on Riemannian manifolds Pennec (2006) that have allowed for their widespread usage in modeling. For example, there exists several PCA generalizations to non-linear data such as the Principal Geodesic Analysis Fletcher et al. (2004), Sommer et al. (2010) and the Geodesic PCA Huckemann et al. (2010). Yet, these methods only use one single tangent space located at the geodesic mean, which can lead them to have significant accuracy error when input data is spread out widely on the manifold. Other algorithms based on kernels Davis et al. (2007) and Markov Chain Monte Carlo sampling Brubaker et al. (2012) have been specifically used in regression tasks on manifolds, but they have not been applied to stochastic modeling problems. There have been recent attempts at removing the tangent space linearization Sommer (2015); Zhang and Fletcher (2013),

[1] http://hi.cs.waseda.ac.jp/~esimo/code/gfmm/

which, however, can not yet scale to the large amounts of data we consider in this work.

Other approaches address classification models on Riemannian manifolds Sanin et al. (2012), Tosato et al. (2010, 2013) and Tuzel et al. (2008). For binary cases, the classifier is usually built in a "flattened" version of the manifold, obtained via the tangent space Tuzel et al. (2008). Multiple category classification problems have been tackled by replacing the tangent space mapping with rolling maps Caseiro et al. (2013), and by using extensions of the Kernel methods to Riemannian manifolds Jayasumana et al. (2013; 2015). In any event, these approaches have been exclusively used for classification problems, which are out of the scope of the current paper, focused on PDF modeling for use as priors.

With regards to density estimation on Riemannian manifolds, various non-parametric approaches Ozakin and Gray (2009), Pelletier (2005) have been proven to be appropriate. However, as their complexity is dependent on the number of training samples, they scale poorly for large datasets. In contrast, semi-parametric models such as the mixture model we propose here can handle large amounts of data efficiently. Dirichlet Processes have been used for fitting mixture models Chang and Fisher (2013), and recently modified to handle the case of the sphere manifold Straub et al. (2015), although, compared to our approach, they have not been extended to arbitrary Riemannian manifolds. Another widely studied manifold is that of tensor fields Lenglet et al. (2006), for which a non-parametric Kernel Density Estimation approach was recently proposed Caseiro et al. (2012). In Muralidharan and Fletcher (2012), individuals on the tangent bundle are modeled and populations are compared with generalized statistical hypothesis tests, but no parametric model is learned. The interesting approach in Archambeau and Verleysen (2005) is similar to ours in spirit, as it proposes a technique to perform Expectation Maximization (EM) on manifolds. However, this work considers data-driven manifolds, resulting in a high computational overhead for large training sets. In addition, it neither estimates the number of clusters, nor makes use of the tangent space, which allows our model to be defined "on the manifold".

As for human pose, it has been traditionally modeled as a tree of connected joints Ionescu et al. (2014), Moeslund and Granum (2001), Moeslund et al. (2006). There have been many different ways of modeling this. One of the most straightforward approaches is to make use of a Gaussian Mixture Model (GMM) Sigal et al. (2004). Another popular trend is to use Gaussian Processes (GP)-based approaches, such as GP-Latent Variable Models Lawrence (2005) and the constrained GP Varol et al. (2012). These have been extended to consider dynamics in the Gaussian process dynamic model (GPDM) Urtasun et al. (2006), Wang et al. (2005), Yao et al. (2011), and also to consider topological constraints Urtasun et al. (2007). Hierarchical variants Lawrence and Moore (2007) (hGPLVM) have also been used in tracking-by-detection Andriluka et al. (2010). However, Gaussian Processes do not scale well to large datasets due to their $\mathcal{O}(n^3)$ complexity for prediction. Sparse approximations do exist Quiñonero-candela et al. (2005), but in general do not perform as well. In contrast, once our model has been estimated it has $\mathcal{O}(1)$ complexity for sampling.

There have been other approaches for modeling human pose such as learning Conditional Restricted Boltzmann Machines (CRBM) Taylor et al. (2010). However, these methods hold on a complex learning procedure that uses several approximations, and make the training of good models harder. Li et al. (2010) proposed the Globally Coordinated Mixture of Factor Analyzers (GCMFA) model which is similar to the GPLVM ones in the sense it is performing a strong non-linear dimensionality reduction. Yet, as GPLVM, it does not scale well to large datasets such as the ones we consider in this work.

We would like to point out that none of the aforementioned approaches is consistent with the manifold of human motion. Some of them use directly the 3D position of the joints while others use angles. In the case of considering 3D points the limb length may vary during the tracking, which is neither realistic nor desirable. In the case of angle representations, they have an inherent periodicity and thus are not a vector space even though they are usually treated as such. Two nearby angles may have very different values, e.g., 0 and 2π. In this case the distance using the angular value would be 2π while the true geodesic distance is 0. Our model can handle both these limitations. Another model that can handle this is the Principal Geodesic Analysis (PGA) Fletcher et al. (2004). However, this model uses a single tangent space and does not model the Probability Density Function (PDF).

Table 1 summarizes the properties of the models we have commented in terms of their complexity, ability to scale, manifold consistence and if they provide or not a PDF. In particular, our approach scales well, is consistent with the manifold, has low complexity (i.e., it just considers a single hyperparameter), and can be easily learned using an Expectation-Maximization algorithm. It is worth noting here that our model is also the fastest of them for sampling (it is $\mathcal{O}(1)$). For instance, our Matlab implementation yields over 100,000 samples per second. An example of our model can be seen in Fig. 2.

3 Geodesic Finite Mixture Model

We next describe our approach, starting with some basic notions on Riemannian geometry and statistics on manifolds. We then integrate these ingredients in a mixture modeling algorithm to build manifold-based generative models.

Table 1 Comparison of several commonly used human pose models

Model	Complexity	Scales?	Consistent?	PDF?
Gaussian diff.	**Low**	**Yes**	No	**Yes**
GMM Sigal et al. (2004)	**Low**	**Yes**	No	**Yes**
PGA Fletcher et al. (2004)	**Low**	**Yes**	**Yes**	No
GPLVM Lawrence (2005)	**Low**	No	No	**Yes**
GPDM Wang et al. (2005)	Medium	No	No	**Yes**
hGPLVM Lawrence and Moore (2007)	Medium	No	No	**Yes**
CRBM Taylor et al. (2010)	High	**Yes**	No	**Yes**
GCMFA Li et al. (2010)	High	No	No	**Yes**
GFMM (Ours)	**Low**	**Yes**	**Yes**	**Yes**

Models are considered to scale if they can handle well large amounts of data ($\sim$ 100K samples) and to be consistent if they use geodesic distances instead of other metrics. The last column reflects whether or not a model is actually modeling the probability density function (PDF) of the data
Favourable properties of the models are highlighted in bold

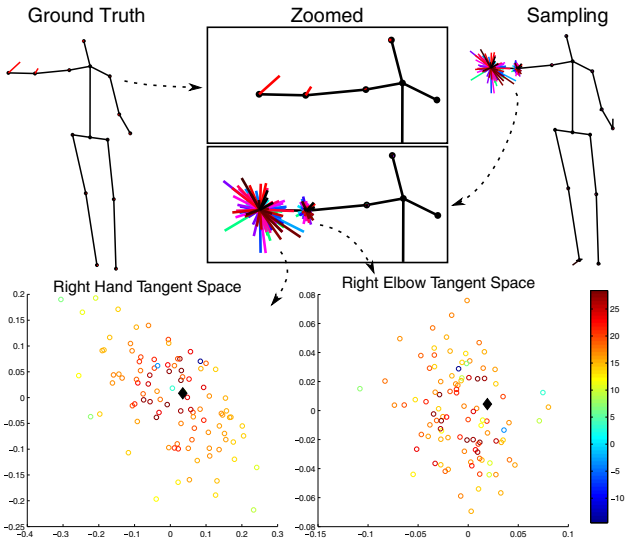


Fig. 2 Example of our motion prior. *Top* for a particular pose, 100 motion samples of the predicted distribution are shown. For visualization purposes the magnitude of the samples is multiplied by 3. *Bottom* visualization of some of the joint samples with their associated log-likelihood. The ground truth is shown with a *black diamond*

3.1 Manifolds, Geodesics and Tangent Spaces

Manifolds arise naturally in many real-world problems. One of the most well-known is the manifold representing spatial rotations. For example, when studying human motion, it is a common practice to use the spatial rotations of the different body parts to obtain a subject-agnostic representation of the whole body pose. This is usually done with angle representations that have an inherent periodicity and thus are not a vector space. By considering the Riemannian manifold of spatial rotations it is possible to use tangent spaces as a local vector space representation, and use powerful statistical tools based on Euclidean metrics. For an in depth description of Riemannian manifolds we refer the reader to Boothby (2003) and Carmo (1992).

A *Riemannian manifold* $(\mathcal{M}, g)$ is a differentiable manifold $\mathcal{M}$ equipped with a metric g, that provides a smooth inner product on the tangent space $T_p\mathcal{M}$ at each point p on the manifold. Consider a parametrized curve $\gamma : [0, 1] \rightarrow \mathcal{M}$ with velocity $\dot{\gamma}(t) = \frac{\partial}{\partial t}\gamma(t)$. A geodesic is a curve that minimizes the distance between the two points $p = \gamma(0)$ and $x = \gamma(1)$. More formally, a geodesic is a curve with null acceleration along M, i.e., the covariant derivative $\frac{D}{\partial t}\dot{\gamma}(t)$ is 0 for all $t \in [0, 1]$. We will denote the length of the geodesic or *geodesic distance* as

$$d(p, x) = \int_0^1 \sqrt{g_\gamma(t)\,(\dot{\gamma}(t),\ \dot{\gamma}(t))}\,dt\ . \tag{1}$$

We can now define the exponential map $\exp_p$ at $p = \gamma(0)$ and its inverse, the logarithmic map $\log_p$ as

$$\exp_p : \begin{array}{ccc} T_p\mathcal{M} & \longrightarrow & \mathcal{M} \\ v & \longmapsto & \exp_p(v) = \gamma(1) = x \end{array}\ ,$$
$$\log_p : \begin{array}{ccc} \mathcal{M} & \longrightarrow & T_p\mathcal{M} \\ x & \longmapsto & \log_p(x) = v \end{array}\ . \tag{2}$$

The exponential map is locally diffeomorphic onto a neighborhood of p. Let $V(p)$ be the largest such neighborhood, then $\log_p(x)$ is defined for any point $x \in V(p)$. Geodesics $\gamma_{(x,v)}(t) = \exp_x(tv)$ from $t = 0$ to infinity can either be minimizing all the way or only up to a time $t_0 < \infty$ and not any further. In this latter case, the point $z = \gamma_{(x,v)}(t_0)$ is called a *cut point*. The set of all cut points forms the *cut locus*, and the corresponding vectors the *tangential cut locus*. The maximal domain of $V(p)$ will be the domain containing 0 and delimited by the tangential cut locus. The geodesic distance can also be written using the logarithmic map as $d(p, x) = \|\log_p(x)\|$ (see Fig. 3).

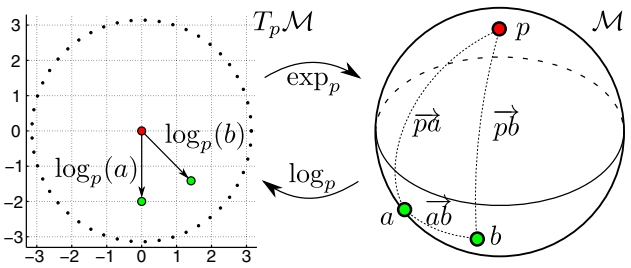


Fig. 3 Representation of geodesics on the S^2 manifold. The tangent space ensures that $\| \log_p(x) \|$ is the true geodesic distance of $\overrightarrow{px}$. However, $\| \log_p(a) - \log_p(b) \|$ is not the geodesic distance of $\overrightarrow{ab}$

In general there is no closed-form of the $\exp_p$ and $\log_p$ maps for an arbitrary manifold. There are, though, approximations for computing them in Riemannian manifolds Dedieu and Nowicki (2005), Sommer et al. (2014). Additionally, efficient closed-form solutions exist for certain manifolds Said et al. (2007). We will discuss some of these manifolds and their associated $\exp_p$ and $\log_p$ maps in the next section.

3.2 Statistics on Tangent Spaces

While it is possible to define distributions on manifolds Pennec (2006), we will focus on approximating Gaussian PDFs of data on a manifold using the tangent space. For instance, the mean of N points x_i on a manifold can be calculated as Karcher (1977):

$$\mu = \arg\min_p \sum_{i=1}^{N} d\,(x_i,\ p)^2 \ . \tag{3}$$

This is iteratively optimized using the $\exp_p$ and $\log_p$ maps,

$$\mu(t+1) = \exp_{\mu(t)}\left(\frac{\delta}{N} \sum_{i=1}^{N} \log_{\mu(t)}\,(x_i) \right) , \tag{4}$$

until $\|\mu(t+1) - \mu(t)\| < \epsilon$ for some threshold ϵ, with δ being the step size parameter.

Knowing the mean value μ and the concentration matrix Γ we can write the distribution that maximizes entropy on the tangent space as a normal distribution centered on the point $\mu \in \mathcal{M}$, corresponding to the origin ($\nu = 0$) in the tangent space:

$$\mathcal{N}_\mu(\nu,\ \Gamma) = a \exp\left(-\frac{\log_\mu(x)^\top \Gamma \log_\mu(x)}{2} \right) , \tag{5}$$

where the normalization constant a and covariance matrix Σ are related to the concentration matrix by:

$$a^{-1} = \int_{\mathcal{M}} \exp\left(-\frac{\log_\mu(x)^\top \Gamma \log_\mu(x)}{2} \right) d\mathcal{M}(x) \ , \tag{6}$$

and

$$\Sigma = a \int_{\mathcal{M}} \log_\mu(x)^\top \log_\mu(x) \exp\left(-\frac{\log_\mu(x)^\top \Gamma \log_\mu(x)}{2} \right) d\mathcal{M}(x) \ . \tag{7}$$

Note that this integral is over the manifold $\mathcal{M}$ and that not all points of the tangent space $T_\mu\mathcal{M}$ correspond to one single point on the manifold, i.e., the tangential cut locus. In particular, for the S^2 sphere, the tangent space is defined inside a circle and not the whole $\mathbb{R}^2$ plane. The circle and the area outside of it forms the tangential cut locus, and for any point on the tangential cut locus there exists more than one minimizing geodesic to the origin. As a simplification, in our formulation we will not consider the tangential cut locus, and will directly approximate normal distributions on the tangent space $T_\mu\mathcal{M}$ at the mean μ with covariance matrix:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} \log_\mu(x_i) \log_\mu(x_i)^\top \ , \tag{8}$$

and

$$a^{-1} = \sqrt{(2\pi)^D \det(\Sigma)} \ , \tag{9}$$

By not taking into account the tangential cut locus we are underestimating the true normalization parameter.

We next perform a simple analysis of the error incurred by obviating the tangential cut locus for the S^2 sphere, which is at a distance π from the origin. We can compute the exact normalization term by calculating the integral of the 2D Gaussian on the tangent space using Eq. (6). We consider the scenario of a normal distribution centered at the origin and with diagonal covariance matrix $\Sigma = \mathrm{diag}(\sigma, \sigma)$. Using polar coordinates we can write:

$$\begin{aligned} (a^*)^{-1} &= \int_0^{2\pi} \int_0^{\pi} \exp\left(\frac{-r^2}{2\sigma} \right) r\, dr\, d\theta \\ &= 2\pi\sigma \left(1 - \exp\left(\frac{-\pi^2}{2\sigma} \right) \right) \ . \end{aligned} \tag{10}$$

If we do not take into account the tangential cut locus, the normalization term of Eq. (9) becomes $a = 2\pi\sigma$. As expected, we are underestimating the true constant by a factor of $a^*/a = 1 - \exp(-\pi^2/(2\sigma))$. To make an estimation error over a 1 %, $\sigma \geq \frac{-\pi^2}{2\log(0.01)} \approx 1.072$. Therefore, unless the distribution has an extremely large covariance, for the S^2 manifold, the estimation error will be less than 1 %. In the

case of human articulations modeled with S^2 joints, most of them do not even have a movement range of 1.072 radians, and their covariance is much smaller, making this error negligible. Furthermore, since we model the data as a mixture of Gaussians, each of the components of the mixture will have a much smaller covariance than the total covariance of the data. Experimentally, we found no difference between considering or not the tangential cut locus.

For an alternate approach to estimate a normal distribution on a Riemannian manifold where the Taylor expansion of the Riemannian metric is used, please refer to Pennec (2006).

3.3 Improving Covariance Estimations

We next discuss alternative approximations to estimate covariance matrices. These new estimates will later be used in the place of the empirical covariance matrix Σ. The standard approach to compute a covariance matrix $S = [s_{jk}]$, defines its entries as:

$$s_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \mu_j)(x_{ik} - \mu_k)^\top , \tag{11}$$

with μ_j and μ_k being the j-th and k-th element of the mean of the N samples x with dimensionality D. Note that this is what was used in Eq. (8).

This empirical covariance matrix is known to be a poor estimation of the true covariance matrix when the number of samples is small compared to their dimensionality, yielding samples which are very sparsely distributed. Several approaches have been proposed for improving this approximation Ledoit and Wolf (2011), Schäfer and Strimmer (2005). In this paper, we will focus on Ledoit-Wolf (LW) Ledoit and Wolf (2004) and Oracle Approximating Shrinkage (OAS) Chen et al. (2010) techniques, as besides being accurate for small datasets, they are also efficient to compute for large training sets. They both belong to the so-called family of "shrinkage estimators".

In linear shrinkage problems, the estimation of the covariance matrix is formulated as a constrained MSE minimization w.r.t. a true covariance Σ:

$$\begin{aligned} &\min_{\rho} E\left[\|\Sigma^* - \Sigma\|_{\mathrm{F}}/D\right] \\ &\text{s.t. } \Sigma^* = \rho\frac{\mathrm{tr}(S)}{D}I + (1-\rho)S , \end{aligned} \tag{12}$$

where $\|A\|_{\mathrm{F}} = \sqrt{\mathrm{tr}(AA^\top)}$ is the Frobenius norm, ρ is the shrinking coefficient, and I is the identity matrix.

Ledoit-Wolf Ledoit and Wolf (2004) proposed using the following shrinking coefficient:

$$\rho_{LW} = \min\left(\frac{\sum_{i=1}^{N}\|x_i x_i^\top - S\|_{\mathrm{F}}/D}{N^2(\mathrm{tr}(S^2) - \mathrm{tr}^2(S)/D)}, 1\right) , \tag{13}$$

which is proven to converge to the optimal solution when $N, D \to \infty$ and $D/N \to c, 0 < c < \infty$.

On the other hand, Chen et al. (2010) propose to use the Oracle Approximating Shrinkage (OAS):

$$\rho_{OAS} = \min\left(\frac{(1-2/D)\mathrm{tr}(S^2) + \mathrm{tr}^2(S)}{(N+1-2/D)\left(\mathrm{tr}(S^2) - \mathrm{tr}^2(S)/D\right)}, 1\right), \tag{14}$$

which is the limiting form of the optimal oracle estimator or ideal value of ρ.

Both the LW and OAS shrinkage estimators have the desirable property that the estimated covariance Σ is in general invertible, unlike the empiric covariance estimation. In the experimental section we will show that they also provide a better covariance estimation for a wide variety of problems.

Another interesting case is when input samples are subject to noise, e.g. due to measurement uncertainty. In this situation we can introduce a prior on the structure of the input data and, for instance, parameterize every sample x_i by a specific Gaussian distribution with covariance Σ_{x_i}:

$$x_i = y_i + \mathcal{N}(0,\ \Sigma_{x_i}^{-1}) , \tag{15}$$

where y_i is the mean value of x_i.

For the sake of completion we will consider each sample x_i to be weighted by w_i (we will use this in the following subsection, for the EM computation). Without loss of generality we assume $\sum_{i=1}^{N} w_i = N$. The mean of the N samples can then be written as:

$$\mu = E[x] = E\left[E[x|y]\right] = E[wy] = \frac{1}{N}\sum_{i=1}^{N} w_i y_i . \tag{16}$$

By using the law of total covariance we can then write the biased weighted sample covariance as

$$\begin{aligned} \Sigma &= \mathrm{cov}(x) = E[\mathrm{cov}(x|y)] + \mathrm{cov}(E[x|y]) \\ &= E[\mathrm{cov}\left(w\mathcal{N}(x,\ \Sigma_x^{-1})\right)] + \mathrm{cov}(wy) \\ &= E[w^2\Sigma_x] + \mathrm{cov}(wy) \\ &= \frac{1}{N}\sum_{i=1}^{N} w_i\left(\Sigma_{x_i} w_i + (y_i - \mu)(y_i - \mu)^\top\right) . \end{aligned} \tag{17}$$

In the results section we will show how it is possible to treat the sample noise Σ_{x_i} as a hyperparameter when estimating mixtures from few samples, and that it behaves as a regularization parameter that helps to improve performance.

3.4 Unsupervised Finite Mixture Modeling

Recall that our ultimate goal is to fit a mixture model on Riemannian manifolds. For this, we will draw inspiration on Figueiredo and Jain (2002), a variant of the EM algorithm Dempster et al. (1977) that uses the Minimum Message Length criterion (MML) Wallace and Freeman (1987) to estimate the number of clusters and their parameters in an unsupervised manner.

Given an input dataset, this algorithm starts by randomly initializing a large number of mixtures. During the Maximization (M) step, a MML criterion is used to annihilate components that are not well supported by the data. In addition, upon EM convergence, the least probable mixture component is also forcibly annihilated and the algorithm continues until a minimum number of components is reached. The approach in Figueiredo and Jain (2002) is designed to work with data in an Euclidean space. To use it in Riemannian manifolds, we modify the M-step as follows.

We define each mixture component with a mean μ_k and a concentration matrix $\Gamma_k = \Sigma_k^{-1}$ as a normal distribution on its own tangent space $T_{\mu_k}\mathcal{M}$:

$$p(x|\theta_k) \approx \mathcal{N}_{\mu_k}\left(0,\ \Sigma_k^{-1}\right) , \tag{18}$$

with $\theta_k = (\mu_k, \Sigma_k^{-1})$. Remember that the mean μ_k is defined on the manifold $\mathcal{M}$, while the concentration matrix Γ_k is defined on the tangent space $T_{\mu_k}\mathcal{M}$ at the mean $\nu_k = 0$. Also note that the dimensionality of the space embedding the manifold is larger than the actual dimension of the manifold, which in turn is equal to the dimension of the tangent space. That is, for an arbitrary embedding of the manifold, $\dim(\text{Embedding}(\mathcal{M})) > \dim(T_p\mathcal{M}) = \dim(\mathcal{M}) = D$. This dimensionality determines the total number of parameters D_θ specifying each component, and, as we will explain below, plays an important role during the component annihilation process. For full covariance matrices it can be easily found that $D_\theta = D + D(D+1)/2$.

We next describe how the EM algorithm is extended from Euclidean to Riemmanian manifolds. For a full derivation of the algorithm please see Appendix 1. Specifically, let us assume that K components survived after iteration $t-1$. Then, in the E-step we compute the *responsibility* that each component k takes for every sample x_i:

$$w_k^{(i)} = \frac{\alpha_k(t-1)\, p(x_i|\theta_k(t-1))}{\sum_{k=1}^{K} \alpha_k(t-1)\, p(x_i|\theta_k(t-1))} , \tag{19}$$

for $k = 1, \ldots, K$ and $i = 1, \ldots, N$, and where $\alpha_k(t-1)$ are the relative weights of each component k.

In the M-step we update the weight α_k, the mean μ_k and covariance Σ_k for each of the components as follows:

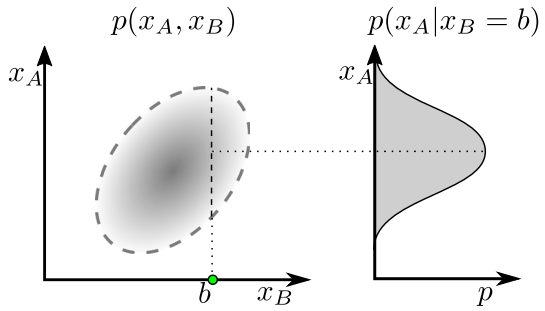


Fig. 4 Representation of a conditioned distribution. In the *left* we show a joint Gaussian distribution over two variables x_A and x_B. We then on the right illustrate the resulting distribution $p(x_A|x_B)$ for a particular point $x_B = b$. We can see that this is also a Gaussian distribution, albeit one-dimensional

$$\begin{aligned}
\alpha_k(t) &= \frac{1}{N}\sum_{i}^{N} w_k^{(i)} = \frac{w_k}{N} , \\
\mu_k(t) &= \arg\min_{p} \sum_{i=1}^{N} d\left(\frac{N}{w_k} w_k^{(i)} x^{(i)},\ p\right)^2 , \\
\Sigma_k(t) &= \frac{1}{w_k}\sum_{i=1}^{N} \left(\log_{\mu_k(t)}(x^{(i)})\right)\left(\log_{\mu_k(t)}(x^{(i)})\right)^{\top} w_k^{(i)} ,
\end{aligned} \tag{20}$$

If we wish to augment the data with noise covariance associated to each sample, it is as simple as adding the weighted average of the noise covariance to $\Sigma_k(t)$ as per Eq. (17).

After each M-step, we follow the same annihilation criterion as in Figueiredo and Jain (2002), and eliminate those components whose accumulated responsibility w_k is below a $D_\theta/2$ threshold. A score for the remaining components based on the Minimum Message Length is then computed. This EM process is repeated until the convergence of the score or until reaching a minimum number of components K_{min}. If this number is not reached, the component with the least responsibility is eliminated (even if it is larger than $D_\theta/2$) and the EM process is repeated. Finally, the configuration with minimum score is retained (see Figueiredo and Jain (2002) for details), yielding a resulting distribution with the form

$$p(x|\theta) = \sum_{k=1}^{K} \alpha_k\, p(x|\theta_k) . \tag{21}$$

3.5 Conditional Distribution

It is possible to use the generative model just described in prediction tasks by estimating the distribution of a subset of variables given another subset of variables. Essentially, given a joint distribution, the distribution of a subset of variables conditioned on another subset of variables is computed as illustrated in Fig. 4. To do this we need to split the dimensions

of the manifold into two subsets, $x = (x_A, x_B)$. Each of these components can be expressed in terms of the mixture as:

$$\theta_k = (\mu_k, \Sigma_k^{-1}) = \left((\mu_{k,A}, \mu_{k,B}), \begin{bmatrix} \Sigma_{k,A} & \Sigma_{k,AB} \\ \Sigma_{k,BA} & \Sigma_{k,B} \end{bmatrix}^{-1} \right) . \tag{22}$$

The conditional distribution of one subset given the other (i.e., the regression function), can be written as:

$$\begin{aligned} p(x_A|x_B, \theta) &= \frac{p(x_A, x_B|\theta)}{p(x_B|\theta_B)} \\ &= \frac{\sum_{k=1}^{K} \alpha_k p(x_B|\theta_{k,B}) p(x_A|x_B, \theta_{k,A})}{\sum_{k=1}^{K} \alpha_k p(x_B|\theta_{k,B})} . \end{aligned} \tag{23}$$

Observe that this is a new mixture model:

$$p(x_A|x_B, \theta) = \sum_{k=1}^{K} \pi_k p(x_A|x_B, \theta_k) , \tag{24}$$

with weights:

$$\pi_k = \frac{\alpha_k p(x_B|\theta_{k,B})}{\sum_{j=1}^{K} \alpha_j p(x_B|\theta_{j,B})} , \tag{25}$$

In the case of the Euclidean space, $p(x_A|x_B, \theta_{k,A})$ can be written as:

$$\begin{aligned} p(x_A|x_B, \theta_{k,A}) &= \mathcal{N}(\mu_{k,A|B}, \ \Sigma_{k,A|B}^{-1}) , \\ \mu_{k,A|B} &= \mu_{k,A} + \Sigma_{k,AB} \Sigma_{k,B}^{-1} \left(x_B - \mu_{k,B} \right) , \\ \Sigma_{k,A|B} &= \Sigma_{k,A} - \Sigma_{k,AB} \Sigma_{k,B}^{-1} \Sigma_{k,BA} . \end{aligned} \tag{26}$$

In our case we assume that the tangent spaces are not being moved and that they are centered on the mean. This allows us to write the conditional probabilities as:

$$\begin{aligned} p(x_A|x_B, \theta_{k,A}) &= \mathcal{N}_{\mu_{k,A|B}}(\nu_{k,A|B}, \ \Sigma_{k,A|B}^{-1}) , \\ \nu_{k,A|B} &= \Sigma_{k,AB} \Sigma_{k,B}^{-1} \log_{\mu_{k,B}}(x_B) , \\ \Sigma_{k,A|B} &= \Sigma_{k,A} - \Sigma_{k,AB} \Sigma_{k,B}^{-1} \Sigma_{k,BA} , \end{aligned} \tag{27}$$

where $\log_{\mu_{k,B}}$ is the subspace of the tangent space $\log_{\mu_k}$ at the subset of the mean $\mu_{k,B}$ for the mixture k. Note that the tangent spaces are not being moved, i.e., they still remain centered on μ, although we are only looking at a subspace of the tangent space.

In the results section we will show how we can use this to predict the kinematics of a person given her/his pose.

3.6 Implementation Considerations

While using the tangent spaces allows representing PDFs of data on manifolds, this comes at a price of higher computational cost as the data must be repeatedly projected back and forth from the tangent space to the manifold. There are, though, several implementation considerations that can be taken into account to improve the efficiency.

For instance, as mentioned in Figueiredo and Jain (2002), we might consider using less expressive covariance matrices (e.g. diagonal ones). However, when using tangent spaces, there is not necessarily a global coordinate frame representation, as the orthonormal basis of the tangent space depends on the $\log_p$ map, and thus, depends on the point p at which it is calculated. When running the EM algorithm, the tangent spaces at step $t+1$ may have a completely different basis than those at step t, and thus, the data likelihood can change drastically, making the optimization much more non-linear. Since (in contrast to full covariance matrices) diagonal matrices can not adapt to arbitrary rotations of the coordinate system, their performance is in general quite poor. This limitation can only be bypassed when there exist a global coordinate frame that can be defined for all tangent spaces independent of the point they are centered on, e.g., the Euclidean spaces $\mathbb{R}^n$.

Nevertheless, when working with a manifold which is the Cartesian product of other manifolds such as $\mathcal{S}_{AB} = \mathcal{S}_A \times \mathcal{S}_B$, it is possible to use a block-diagonal matrix of the form:

$$\Sigma_{\mathcal{S}_{AB}} = \begin{pmatrix} \Sigma_{\mathcal{S}_A} & 0 \\ 0 & \Sigma_{\mathcal{S}_B} \end{pmatrix} = \begin{pmatrix} \Gamma_{\mathcal{S}_A}^{-1} & 0 \\ 0 & \Gamma_{\mathcal{S}_B}^{-1} \end{pmatrix} = \Gamma_{\mathcal{S}_{AB}}^{-1} , \tag{28}$$

which by construction is a valid covariance matrix and avoids the issue of dealing with arbitrary orthonormal basis. The null row and column elements highly simplify the computational cost. By using less expressive covariance matrices, the model has fewer degrees of freedom and generally converges in fewer iterations, besides requiring less training data.

Furthermore, while in the Euclidean case the mean of a set of points can be computed in closed form, when working with manifolds we need to do this iteratively. In our implementation this is required in the M-step, where the parameters θ_k are estimated as a weighted combination of terms. By considering only a subset of samples S such that $\sum_{i \in S} w_k^{(i)} > \epsilon_s$ for a certain threshold ϵ_s, it is possible to improve the computational efficiency without sacrificing accuracy.

In order to improve the initialization, we consider using the k-means algorithm in the embedding space as a coarse initialization for the algorithm. This ensures a fair spread of the initial clusters over the data in contrast to a random sampling that may fail when the data is unevenly distributed. In the results section we will show that manifolds with large

dimension lead to fewer clusters being annihilated in the initial iteration and more stable convergence properties.

4 Manifolds

We will now describe several manifolds that we will use in the experimental section. For each manifold we will briefly discuss their structure and the $\exp_p$ and $\log_p$ map implementations.

4.1 Quadratic Surfaces

In general, there is no closed-form of the $\exp_p$ and $\log_p$ maps for an arbitrary Riemannian manifold. There are, though, approximations for computing them Dedieu and Nowicki (2005), Sommer et al. (2014). In particular we will consider implicitly defined surfaces.

Computing the $\exp_p$ map corresponds to solving an initial value ordinary differential equation problem. Refer to Dedieu and Nowicki (2005) for a neat numerical algorithm to obtain solutions to the map.

The computation of the $\log_p$ map is harder and is based on a shooting algorithm. This relies on the $\exp_p$ map to iteratively refine an initial guess, and is a natural generalization of error correction from the Euclidean space to manifolds. We use the implementation proposed in Sommer et al. (2014).

4.2 S^2 Manifold

There is an explicit mapping between the unit sphere S^2 and its tangent space T_pS^2 Said et al. (2007). Let $x = (x_1, x_2, x_3)^\top$, $y = (y_1, y_2, y_3)^\top$ be two unit spoke directions in S^2 and $v = (v_1, v_2)^\top$ a point in T_pS^2. The $\exp_p$ and $\log_p$ maps are in this case:

$$\begin{aligned} \exp_p(v) &= R_p^{-1}\left(v_1\frac{\sin\|v\|}{\|v\|},\ v_2\frac{\sin\|v\|}{\|v\|},\ \cos\|v\|\right), \\ \log_p(x) &= \left(y_1\frac{\theta}{\sin\theta},\ y_2\frac{\theta}{\sin\theta}\right), \end{aligned} \tag{29}$$

where R_p is the rotation of p to the north pole, $\|v\| = (v_1^2 + v_2^2)^{\frac{1}{2}}$, $y = R_p x$ and $\theta = \arccos(y_3)$.

4.3 Human Pose Manifold

The human pose is commonly represented using a discrete set of points in 3D space that correspond to different articulations Ionescu et al. (2011), Simo-Serra et al. (2013; 2012). For a specific individual, the distance between two consecutive joints, e.g., elbow and hand, is fixed. It is therefore common to separate the specific characteristics of the individual given by the distances between two consecutive joints, from his/her pose, i.e., the relative rotation between two consecutive joints Ionescu et al. (2014). We will therefore consider the human pose as the set of relative rotations between all pairs of consecutive joints which forms a tree structure Sommer et al. (2010), Tournier et al. (2009). We will further represent this relative motion as points on a sphere. That is, given a specific joint, the next consecutive joint will lay on the S^2 sphere centered on the previous one. The whole pose will thus be represented as the Cartesian product of all the relative rotations between consecutive joints. We will write the human pose manifold $\mathcal{H}$ as

$$\mathcal{H} = S^2 \times S^2 \times \cdots \times S^2 . \tag{30}$$

In this case the $\exp_p$ and the $\log_p$ maps for the human pose manifold will consist of the Cartesian product of the $\exp_p$ and $\log_p$ maps for all consecutive joints, which is one less than the total number of joints. Note that while there exist other manifolds for 3D human pose, such as one defined by using forward kinematics Hauberg et al. (2012), they do not have closed form solutions for the $\exp_p$ and $\log_p$ operators.

4.4 Joint Pose and Kinematic Manifold

We can obtain a joint human pose and kinematics manifold with the tangent bundle of the human pose manifold, which we equip with a Riemannian metric called the Sasaki metric Sasaki (1958). The tangent bundle is defined as:

$$T\mathcal{M} = \{(x, v) \mid x \in \mathcal{M},\ v \in T_x\mathcal{M}\} . \tag{31}$$

Let (u, w) be a vector tangent to $T\mathcal{M}$ at a point (x, v). Both u and w must be lifted from the tangent space $T_x\mathcal{M}$ to $T_{(x,v)}T\mathcal{M}$. The lift of the u component is called the *horizontal lift* of u and denoted u^h. Likewise, the lift of the w component is called the *vertical lift* of w and denoted w^v. Geodesics along u^h move x while parallelly translating u, whereas geodesics along w^v move v linearly while keeping x fixed. Given two elements $a = (x_1, v_1), b = (x_2, v_2) \in T\mathcal{M}$, the Sasaki metric $\hat{g}(a, b)$ is given as

$$\begin{aligned} &\hat{g}(x_1^h, x_2^h) = g(x_1, x_2) , \\ &\hat{g}(x_1^h, v_2^h) = \hat{g}(x_2^h, v_1^h) = 0 , \\ &\hat{g}(v_1^h, v_2^h) = g(v_1, v_2) , \end{aligned} \tag{32}$$

where g is the metric on $\mathcal{M}$. Notice that there are no cross-terms between x and v.

For two consecutive poses x_1 and x_2 acquired at a constant framerate, we can compute the velocity v_{12} between x_2 and x_1 directly on the tangent space through the logarithmic map at x_1 and thus define the joint pose and kinematic manifold as:

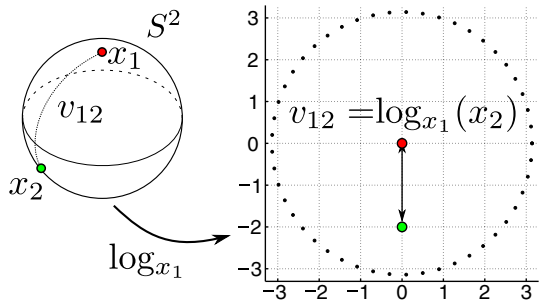


Fig. 5 Visualization of the velocity. Velocities correspond to points on the tangent space at x_1. Given a point x_2 (corresponding to the pose acquired right after x_1), the velocity is the curve going from x_1 to x_2 which is equivalent to a *straight line* in the tangent space. The Euclidean norm of v_{12} on the tangent space corresponds to the geodesic distance from x_1 to x_2

$$T\mathcal{H} = \left\{(x_1, v_{12}) = \left(x_1, \log_{x_1}(x_2)\right) \mid x_1, x_2 \in \mathcal{M}\right\} , \quad (33)$$

where $\|v_{12}\|$ is the geodesic distance between both points as shown in Fig. 5.

We next define the exponential and logarithmic maps for the elements $(x, v) \in T\mathcal{H}$. For a local neighbourhood of x, the elements $v \in T_x\mathcal{H}$ form a vector space and thus the operators can be simply defined as in the Euclidean case with

$$\log_{v_1}(v_2) = v_2 - v_1 \quad \text{and} \quad \exp_{v_1}(v_2) = v_2 + v_1 . \quad (34)$$

This ensures that the mean of the data on the tangent space at the geodesic mean will be 0, i.e, the mean of $\{\log_p(x_i)\}_{x_i}$ will be 0 if p is the geodesic mean.

We can also further extend this manifold by including more poses acquired sequentially at a constant frame rate. For example, given three consecutive poses x_1, x_2 and x_3 it is possible to map these points to a manifold in which one pose is the reference and the other two poses are considered offsets from that reference or tangent vectors. That is (x_1, x_2, x_3) is mapped to $(\log_{x_2}(x_1), x_2, \log_{x_2}(x_3))$ where x_2 would be the local reference.

5 Results

In this section we will extensively evaluate our model on both synthetic and real data. We consider a number of manifolds and evaluation procedures to show the flexibility and diversity of our model. The section is structured as follows:

1. Evaluation on the recovery of synthetic distributions on quadratic surfaces.
2. In depth results on modeling synthetic distributions on the S^2 sphere.
3. Modeling human pose and kinematics on the large-scale Human3.6M dataset Ionescu et al. (2014).
4. Tracking prior by extending the manifold with its tangent bundle.

5.1 Recovering Distributions on Quadratic Surfaces

We first present experiments on two 2D manifolds defined by

$$\mathcal{M} = \left\{(x, y, z) \mid cx^2 + y^2 + z^2 = 1\right\} . \quad (35)$$

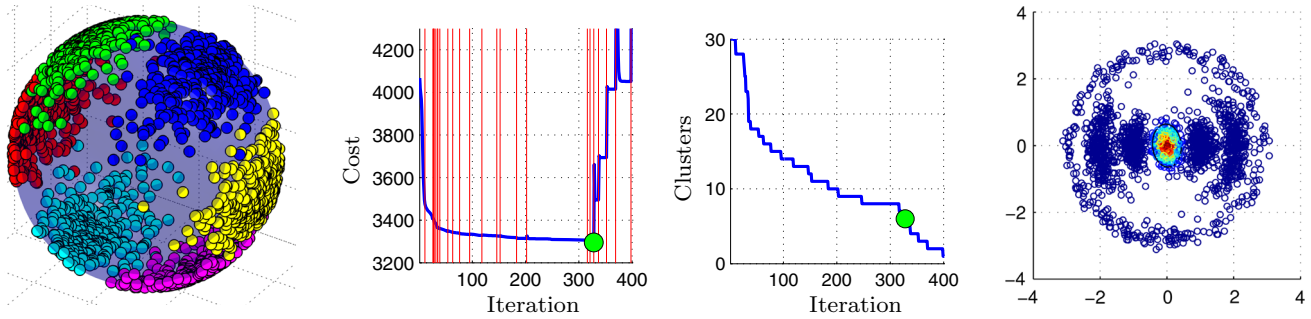
For the first example we generate 1800 points on the sphere S^2 (i.e. $c = 1$) using 6 clusters with parameters $\mu_i = [\cos(i\pi/3), 0, \sin(i\pi)/3]$ and $\Sigma_i = \text{diag}(0.2, 0.3)$ for $i = 1, \ldots, 6$. The algorithm is initialized with 30 clusters. This manifold has the closed-form solution for the $\exp_p$ and $\log_p$ operators given by Eq. (29), allowing the method to execute in under a minute. Fig. 6 shows how the final solution retrieves the 6 clusters used to generate the data.

For the second example we generate 1500 points from a mixture of 5 Gaussians on the manifold of Eq. (35) with $c = -2$, as shown in Fig. 7a. The Gaussian parameters used in this case are:

$$\begin{array}{ll} \mu_1 = [0.83, -1.09, 1.09] & \Sigma_1 = \text{diag}(0.20, 0.30) \\ \mu_2 = [0, 0, 1] & \Sigma_2 = \text{diag}(0.25, 0.10) \\ \mu_3 = [0.30, -0.77, 0.77] & \Sigma_3 = \text{diag}(0.20, 0.10) \\ \mu_4 = [0, 0, 1] & \Sigma_4 = \text{diag}(0.10, 0.25) \\ \mu_5 = [0, -0.89, 0.44] & \Sigma_5 = \text{diag}(0.25, 0.10) \end{array}$$

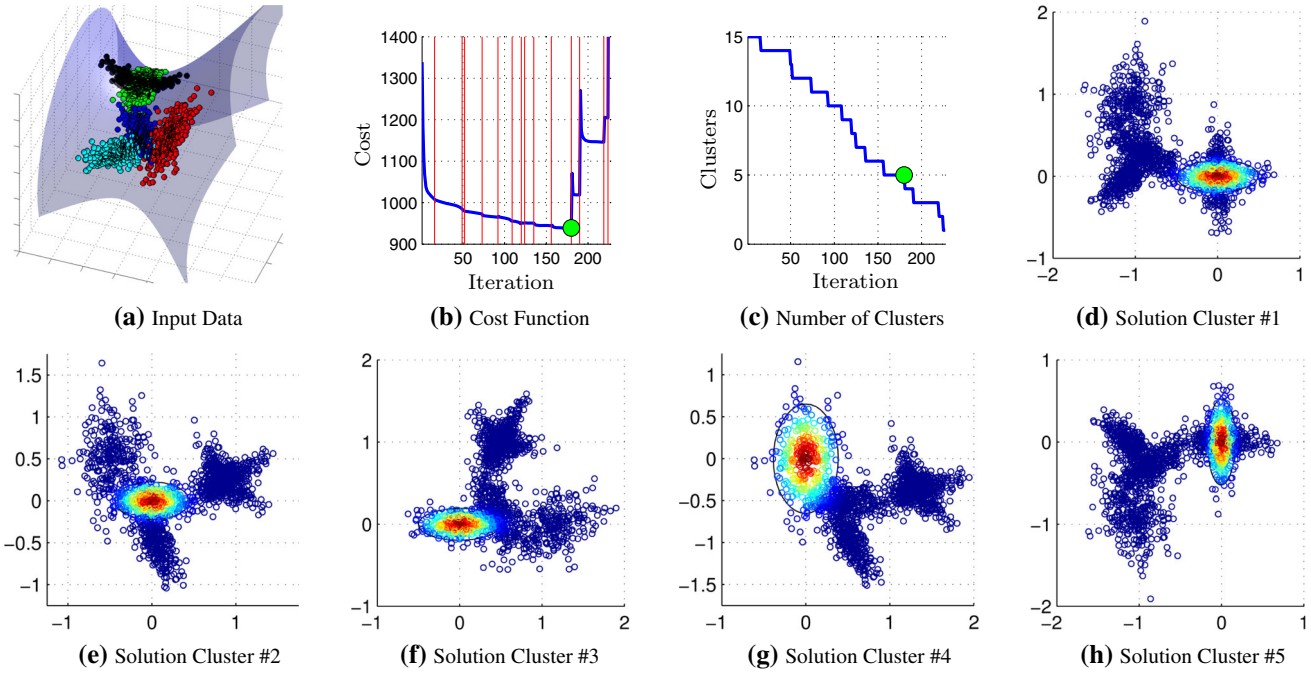
It is worth to highlight two important details about this mixture. First, the covariances depend on the local orthonormal basis of the tangent plane, thus even if they are diagonal, in practice they are not when projected back onto the manifold, as shown in Fig. 7a. Second, clusters 2 and 4 share the same mean. In this example, the algorithm is initialized with 15 clusters and uses the generic forms of the $\exp_p$ and $\log_p$ operators that rely on the derivative of the implicit manifold equation as detailed in Dedieu and Nowicki (2005), Sommer et al. (2014). Additionally, the threshold ϵ_s described in Sect. 3.6 is set to 0.999 to speed up the computations. In this scenario the underlying distribution is recovered as shown in Fig. 7.

In these synthetic experiments, we also analyze the effects of the non-linearities of the $\exp_p$ and $\log_p$ operators by evaluating the method on a sphere (Eq. (35) with $c = 1$) using 6 clusters with mean and covariance parameters $\mu_i = [\cos(i\pi/3), 0, \sin(i\pi)/3]$ and $\Sigma_i = \text{diag}(\sigma, \sigma)$ for $i = 1, \ldots, 6$, and for increasing values of σ. Several examples of input distributions with different covariances are shown in Fig. 8a. The effect of the number of input samples is seen by testing with $N = \{600, 1800, 6000\}$. The algorithm parameters used are the same as in the aforementioned sphere example.



(a) Input Data **(b)** Cost Function **(c)** Number of Clusters **(d)** Solution Cluster #1

Fig. 6 Sphere example. **a** Input data on the manifold. It consists of 1800 points sampled from a mixture of 6 Gaussians. Note that they are colored only for visualization purposes; our algorithm does not know a priori these clusters. **b** Evolution of the cost function, based on the minimum message length of all components of the mixture. *Vertical lines* represent iterations in which a cluster is annihilated. The optimal mixture (with 6 components) is highlighted with a *green dot*. **c** Evolution of the number of clusters. **d** The points projected onto the tangent space of one specific cluster from the solution mixture. Each point is colored by the value of Eq. (18). The cluster on the opposite side of the point the tangent space is centered on is seen to be spread around the cut locus, which is a *circle* of radius π. Best viewed in *color* (Color figure online)



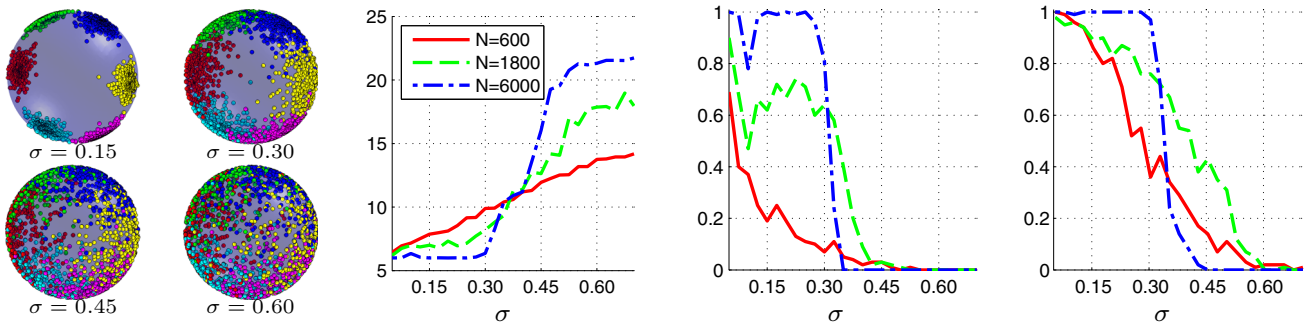
(a) Input Data **(b)** Cost Function **(c)** Number of Clusters **(d)** Solution Cluster #1

(e) Solution Cluster #2 **(f)** Solution Cluster #3 **(g)** Solution Cluster #4 **(h)** Solution Cluster #5

Fig. 7 Quadratic surface example. **a** Section of the manifold with input data generated from 1500 points sampled from a mixture of 5 Gaussian distributions. **b**, **c** refer to Fig. 6. All retrieved clusters are shown in (**d**–**h**). Best viewed in *color* (Color figure online)

The results are shown in Fig. 8b–d. With less spread Gaussians and little overlap, having more data can be seen to be beneficial. However, with more overlap and samples, generally the data gets partitioned into more clusters. These results seem to indicate that the algorithm tends to implicitly favor smaller Gaussians over larger ones, suggesting that there shouldn't be problems with approximating distributions. It is also worth mentioning that these results are for clustering. When estimating a probability density function, the number of clusters is not particularly important as long as the underlying density is properly approximated.

Finally, in order to evaluate the benefit of using multiple tangent spaces over a single one, we perform a comparison on the sphere manifold, in two situations: the same 6 clusters as in Fig. 8 with $\Sigma_i = \text{diag}(0.2, 0.3)$, and when fully covering the sphere with two additional clusters centered at $(0, 1, 0)$ and $(0, -1, 0)$. We also compare against an approach that uses von Mises–Fisher (vMF) distributions Banerjee et al. (2005), which is specifically designed to cluster data on a sphere, and two approaches based on Dirichlet Processes (DP-GMM and DP-TGMM). DP-GMM uses split/merge proposals with parallel sampling in order to estimate a mix-



(a) Example Input Data **(b)** Mean Clusters **(c)** $K = 6$ Solutions **(d)** $K \leq 9$ Solutions

Fig. 8 Effect of cluster size and number of samples. Evaluation of the proposed algorithm for increasing covariance sizes diag(σ, σ) and three amounts of samples points N on the sphere S^2. For each combination of these parameters we run 100 different experiments and we report the average results. **a** Examples of the different input data distributions we consider. **b** Mean number of solution clusters found. **c**, **d** Ratio of estimated solutions with a number of clusters subject to different constraints. Best viewed in *color* (Color figure online)

Table 2 Recovering distributions on the Sphere manifold

	6 Clusters		8 Clusters	
	Clusters	Correct	Clusters	Correct
Ours	6.09 (0.32)	0.92	8.00 (0.00)	1.00
1-TM	7.05 (1.38)	0.46	15.25 (2.17)	0.00
vMF	16.59 (1.71)	0.00	19.86 (2.35)	0.00
DP-GMM Chang and Fisher (2013)	7.83 (1.27)	0.08	3.43 (3.26)	0.26
DP-TGMM Straub et al. (2015)	4.49 (1.98)	0.55	8.34 (1.02)	0.51

We show results on recovering an original distribution on the Sphere manifold with 6 and 8 clusters. Results obtained from 100 different evaluations and 1000 samples per cluster. We compare our method with using a single tangent space (1-TM), von Mises–Fisher distributions (vMF), and two variants of dirichlet process mixture models (DP-GMM and DP-TGMM)

ture model. This was then extended to sphere manifolds to use multiple tangent spaces (DP-TGMM). For the vMF distributions we use our own improved implementation based on Figueiredo and Jain (2002) (see Appendix 2), while we use the authors implementation of DP-GMM and DP-TGMM from Straub et al. (2015). We use the default parameters for all approaches except DP-GMM and DP-TGMM in which we augment the number of allowed iterations to improve their results.

The results are summarized in Table 2. In the 6-cluster case our algorithm retrieves the correct number of clusters in a 92 % of the experiments, while one single tangent plane only provides a 46 % of success. Note that we evaluate the performance of the methods based only on the number of clusters, and not comparing the entire density probability. In the following subsection we will show that the distributions obtained with one single tangent plane are also much less representative as those obtained with the proposed approach. In the 8-cluster case our algorithm's performance improves and it always finds the correct clusters, while a single tangent space always fails, with an average of 15 clusters found. This is likely caused by the fact that the 8-clusters are evenly distributed around the sphere causing a single tangent space to suffer from extreme deformation. This amount of deformation on the contrary helps our optimization process to place mixtures, and thus tangent spaces around the sphere exploring more of the solution space and finding the near optimal solution. On the other hand, the 6-clusters do not have such a large amount of deformation and our approach is sometimes unable to properly locate the cluster means. Using vMF distributions results in an oversegmentation of the data in both experiments. This is due to the fact that the vMF distributions use a single parameter κ for the concentration of the data, while our model allows for much more expressive covariances in the form of matrices. The Dirichlet process-based approaches show promising results, especially the sphere-specific approach DP-TGMM, which is able to find the correct distribution roughly half of the times. However, in all cases this approach is outperformed by our approach. These results clearly show the advantage of using multiple tangent planes to better approximate manifold distributions. In the next subsection we will evaluate more into detail our method on this manifold.

5.2 Estimating Distributions on the S^2 Sphere

In order to evaluate the different hyperparameters of our model, we perform much more in depth analysis of clustering on the S^2 sphere manifold. In particular, we focus on the influence of the number of samples used to learn the distribution, the specific covariance estimation algorithm, k-means initialization, and the sample noise. We consider a S^2 sphere with a synthetic distribution formed by 6 clusters as shown in Fig. 6a. As a metric we will consider the log-likelihood of 60,000 test samples, randomly generated from the distribution (10,000 samples per cluster). That is, for each test sample x we first compute $l(x) = \log(\sum_k \alpha_k p(x|\Theta_k))$, and then we report the average log-likelihood for all samples. The larger this value, the better the samples fit the estimated distribution.

Additionally, unless otherwise specified, for the rest of parameters we use the same values as in the previous subsection. For all experiments, we randomly sample from the distributions 100 times to obtain different training sets and report the mean and standard deviation of the results for all 100 learned models, each evaluated on the 60,000 test samples.

In our first experiment we simultaneously considered the effect of the number of training samples, type of covariance estimator and whether or not k-means is used for initialization. We report the results in Table 3. We can see that the shrinkage-based estimators outperform the empirical-based ones, especially when the number of training samples is small compared to the dimensionality of the manifold. Also note that our approach consistently outperforms methods relying on a single tangent space and the approach using von Mises–Fisher distributions. Regarding the use of k-means for initialization, we did not observe that much difference, likely due by the low-dimensionality of the manifold. Finally, no matter the approach used, performance degrades with decreased amounts of training data.

We also looked into the effect of sample noise when clustering. Even though the true sample noise is not known, there are applications in which it may be possible to obtain it, such as when clustering data obtained from sensors with known properties. In particular we look at the extreme case of small amounts of training data given the manifold dimensionality. We summarize the results of this analysis in Table 4. Note that all approaches in this case benefit from this added sample noise until it becomes too large. The best result is obtained with Ledoit-Wolf covariance shrinkage approach and $\Sigma_{x_i} = 10^{-2}$ sample noise (the units of this noise could be interpreted as an angle in radians).

Table 3 Comparison of the effect of number of training samples, k-means initialization, and different covariance estimators

		Ours						1-TM	vMF
	Cov. Est.	Empirical		LW Ledoit and Wolf (2004)		OAS Chen et al. (2010)		Empirical	vMF
	k-Means	No	Yes	No	Yes	No	Yes	No	No
6000 training samples	Mean	−0.7474	−0.7474	**−0.7466**	**−0.7466**	−0.7473	−0.7472	−10.9331	−1.7722
	SD	0.0044	0.0041	**0.0035**	**0.0035**	0.0041	0.0043	0.6564	0.0040
600 training samples	Mean	−0.8092	−0.8097	**−0.7773**	−0.7778	−0.8012	−0.7953	−10.3445	−1.8386
	SD	0.0279	0.0288	**0.0127**	0.0128	0.0239	0.0214	0.8049	0.0145
60 training samples	Mean	−1.4820	−1.4826	−1.1064	**−1.1031**	−1.1829	−1.1849	−10.0075	−2.2944
	SD	0.3094	0.2795	0.1121	**0.1095**	0.1702	0.1434	2.0754	0.1662

Evaluation of various settings on a synthetic distribution on the S^2 sphere using the log-likelihood of 60,000 random test samples. The best result for each fixed number of training samples is highlighted in bold

Table 4 Comparison of the effect of the sample noise, k-means initialization, and different covariance estimators

Sample noise	Cov. Est.	Empirical		LW Ledoit and Wolf (2004)		OAS Chen et al. (2010)	
	k-Means	No	Yes	No	Yes	No	Yes
0	Mean	−1.4820	−1.4826	−1.1064	−1.1031	−1.1829	−1.1849
	SD	0.3094	0.2795	0.1121	0.1095	0.1702	0.1434
10^{-3}	Mean	−1.3673	−1.4093	−1.0890	−1.0811	−1.1473	−1.1572
	SD	0.2038	0.2529	0.1071	0.1050	0.1270	0.1423
10^{-2}	Mean	−1.0877	−1.0916	−1.0265	−1.0157	−1.0138	**−1.0057**
	SD	0.0987	0.0979	0.0730	0.0609	0.0781	**0.0772**
10^{-1}	Mean	−1.2261	−1.2287	−1.2374	−1.2440	−1.2595	−1.2621
	SD	0.0395	0.0399	0.0437	0.0440	0.0432	0.0437

Evaluation on a synthetic distribution on the S^2 sphere with 60 training samples (10 per cluster) using the log-likelihood of 60,000 random test samples. The best result is shown in bold

5.3 Human Pose Prior

To illustrate a practical utility of our approach, we used it to model human pose on the Human3.6M Dataset Dataset Ionescu et al. (2011; 2014) which has different subjects performing a variety of activities. We consider a simplified model of the human body with 15 joints, represented in a 24-dimensional pose manifold. The corresponding block-diagonal covariance (as in Simo-Serra et al. 2014) has 46 non-zero elements, and the full covariance matrix has 576 non-zero elements.

We split the dataset by using all the 15 categories of actions, each comprising two subcategories, for actors 5, 6, 7, 8, 9, and 11 for the training set, and use actor 1 as the test set. We perform an in-depth analysis of our model using this split, and finally show results on generalization using other actors. The diversity of the actions makes the dataset very challenging to learn. This gives us 465,325 frames for training and 62,064 frames for testing. Since the frames are highly correlated because motions are smooth, we perform a random subsampling before training our model. To assess its influence, we will consider four different subsampling levels.

We consider three baselines: standard Gaussian mixture model directly on 3D joint positions, using a single tangent space for clustering (1-TM), and clustering with von Mises–Fisher distributions (vMF). We also compare five different variants of our model. The block-diagonal covariance matrix approach of Simo-Serra et al. (2014), without covariance shrinkage estimators and with them. Furthermore, we observed that when not using block matrices, the empirical covariance estimation run into numerical issues, while shrinkage estimators performed stable. Therefore, we also report results for non-block-diagonal covariance matrices. All these cases are summarized in Table 5.

Results show that, as expected, the non-manifold based approach (GMM) fails. The one tangent plane model (1-TM) performs poorly due to the non-linearities of the approximation, which gets worse with more training data. The von Mises–Fisher performs consistently better. We see that our models perform best, although with too much heavily correlated training data (larger subsampling ratios) they tend to suffer from overfitting and performance degrades. In particular, the best performance is obtained using the Oracle Approximating Shrinkage algorithm for covariance estimation with 5 % of the training data. This model greatly outperforms the previous work and confirms the importance of more robust covariance estimation algorithms.

Table 5 Modeling 3D human pose

	Model	Subsampling			
		0.01	0.05	0.15	0.30
	GMM	−708.4	−708.4	−708.4	−708.4
	1-TM	−1.653	−2.647	−3.959	−3.975
	vMF	3.413	3.414	3.412	3.416
Ours	Block	4.995	4.570	3.130	1.388
	LW block	5.285	5.581	3.614	1.560
	OAS block	4.937	4.621	3.076	1.289
	LW	7.272	8.039	5.225	2.765
	OAS	7.250	**8.411**	7.276	4.624

Log-likelihood of test data for various models. Evaluating models learned with different subsamplings of the training data. We consider Gaussian mixture models (GMM), using a single tangent plane (1-TM), von Mises–Fisher distributions (vMF), and five different variants of our model (using block covariance matrices or not, and what type of covariance estimation algorithm is used). The best result is highlighted in bold

5.4 Tangent Bundle-Based Tracking Prior

We also evaluate the proposed algorithm in a tracking task, in which, given several consecutive frames we seek to predict the next one. We will consider three manifolds:

Table 6 Log-likelihood of joint pose and kinematic models $p(\mathcal{V})$

Manifold	Subsampling	0.05			0.15			0.30		
	Method	#Mix	Train	Test	#Mix	Train	Test	#Mix	Train	Test
$\mathcal{V}_1$	Block	38	18.708	16.767	173	24.232	17.281	365	28.276	17.030
	LW block	25	16.347	15.197	161	22.417	16.991	335	26.486	17.251
	OAS block	40	18.701	16.773	177	24.382	17.502	370	28.318	16.935
	LW	–	–	–	9	24.036	20.087	22	28.267	21.711
	OAS	–	–	–	8	25.976	22.134	22	31.842	24.105
$\mathcal{V}_2$	Block	19	25.874	30.249	75	32.447	34.694	180	36.545	35.277
	LW block	17	24.522	28.982	72	31.348	33.812	170	35.313	34.834
	OAS block	17	25.184	29.587	78	32.669	34.807	167	36.078	35.133
$\mathcal{V}_3$	Block	13	17.825	27.294	43	26.575	34.819	98	31.383	37.489
	LW block	11	16.013	25.106	37	24.759	33.004	84	29.565	36.478
	OAS block	11	16.772	25.106	47	27.075	34.959	98	31.173	37.085

We evaluate on several different manifolds for various degrees of subsampling of the training data. For each case, we plot the number of estimated mixture components (#Mix), and the log-likelihood values for the train and test sets. Testing is performed on subject 1 while training is performed on the rest of the subjects

$$\begin{aligned}
\mathcal{V}_1 &= \left(x_t, \log_{x_t}(x_{t+1})\right) = T\mathcal{H}\,, \\
\mathcal{V}_2 &= \left(\log_{x_t}(x_{t-1}), x_t, \log_{x_t}(x_{t+1})\right)\,, \\
\mathcal{V}_3 &= \left(\log_{x_t}(x_{t-2}), \log_{x_t}(x_{t-1}), x_t, \log_{x_t}(x_{t+1})\right)\,, \qquad (36)
\end{aligned}$$

where x_t is the pose at frame t.

For each manifold we will estimate a mixture that learns $p(\mathcal{V})$. Once this is learned it is then possible to predict the next pose by using the conditional distribution $p(\log_{x_t}(x_{t+1})|\mathcal{V}^*)$ where $\mathcal{V}^*$ is the manifold resulting from removing $\log_{x_t}(x_{t+1})$ from $\mathcal{V}$. Recall that this marginal is indeed another mixture model.

We first perform a quantitative evaluation by looking at the log-likelihood of $p(\mathcal{V})$ for the test and train sets. As we did in the previous subsection, we subsample the training data to gain in efficiency. This is possible with no performance loss, due to the large degree of redundancy in the training data. A subsampling percentage of 15 % corresponds to 69,799 training samples, roughly the same number as the test set. The parameters are set to the same values as in the pose-only case. Table 6 summarizes the results, in which for each method and experimental setup, we display the number of components of the estimated mixture and the log-likelihood of the train and test sets. Note that the number of estimated components increases with the amount of data. As the dimension of the manifold is increased, it is harder to learn the models due to the additional degrees of freedom of each covariance matrix. It is important to note that we are unable to use full covariance matrices on the $\mathcal{V}_2$ and $\mathcal{V}_3$ manifolds due to the large number of degrees of freedom (5184 and 9216 respectively), while it is possible to use them with 0.15 and 0.30 subsampling on the $\mathcal{V}_1$ manifold (2304 degrees of freedom). Furthermore, it is only possible to estimate the covariance of such large matrices with shrinkage covariance estimators. Using a full covariance matrix provides a large increase in performance over the block-diagonal when applicable. Additionally, adding more temporal information increases performance, although, more data is needed to learn these models.

We next evaluate the model to predict future positions, that is, the log-likelihood of $p(\log_{x_t}(x_{t+1})|\mathcal{V}^*)$, instead of the global likelihood $p(\mathcal{V})$. We compare against a GD approach both trained on a global level (a single Gaussian is averaged for all joints) and on a local level (a single Gaussian is averaged for each joint independently).[2] Recall that both these approaches consists of simply defining the motion as a Gaussian distribution centered on the previous frame, and they operate directly on the Euclidean space, and not on the manifold. We train several kinematic models with different degrees of subsampling of the training data, and report the results in Table 7. We can see that the local Gaussian diffusion (LGD) model outperforms the standard Gaussian diffusion (GD) model. Yet, our model outperforms both of them by a considerable margin. These results largely agree with Table 6.

In order to assess the generalization capability of the algorithm, we evaluate our approach with different subject splits and summarize the results in Table 8. We use the leave-one-out strategy: using all the subjects except one for training, which is used for testing. We evaluate as many times as there are subjects, changing the subject which is being left out for testing each time. For fairness with the GD approaches that only account for pose (and not velocity) information, we just

[2] Since vMF-distributions are not directly applicable to predicting velocities on the hypersphere we do not include them in this experiment.

Table 7 Evaluation of velocity estimation

Manifold	Method	Subsampling		
		0.05	0.15	0.30
–	GD	5.450	5.425	5.431
–	LGD	6.428	6.412	6.410
$\mathcal{V}_1$	Block	11.842	11.739	11.493
	LW block	10.503	11.097	10.995
	OAS block	11.713	11.829	11.363
	LW	–	12.115	12.937
	OAS	–	13.474	14.100
$\mathcal{V}_2$	Block	14.715	16.501	16.819
	LW block	14.008	15.993	16.402
	OAS block	14.398	16.644	16.725
$\mathcal{V}_3$	Block	14.024	16.570	17.341
	LW block	13.445	16.053	17.050
	OAS block	13.591	16.517	17.259

$p(\log_{x_t}(x_{t+1})|\mathcal{V}^*)$. Evaluation on different manifolds and subsampling levels of the training data. We report the log-likelihood of the velocity prediction of the testing set. Testing is done on subject 1 while training is done on the rest of the subjects. Gaussian diffusion (GD) and local gaussian diffusion (LGD) are used as baselines. Both approaches operate on the Euclidean space and not on the manifold

consider the $\mathcal{V}_1$ manifold. For all approaches a 0.15 subsampling ratio is used. Regarding our approach, we use it both with block diagonal matrices, as in Simo-Serra et al. (2015), and with the improved version based on the Oracle approximating shrinkage (OAS). Observe that the latter yields a large performance gain. On average, both alternatives outperform the Gaussian Diffusion baselines, except for subject 7. In this dataset in particular, the actors were given a lot of freedom to perform the actions. It is likely that subject's 7 motion largely deviates from other subjects. It is also interesting to note that subjects 1, 8, and 11 have better performance on the test set rather than the train set. This is likely due to the fact that there is correlation across subjects.

Finally, we show some qualitative examples in Fig. 9. For this, we directly sample from the conditional distribution for several frames. It is worth noting that we can obtain 100,000 samples in 0.85 seconds on a Intel Core i7 2.93GHz CPU using a Matlab implementation.

6 Conclusions

We have presented a novel data-driven approach for modeling the probability density function of data located on a Riemannian manifold. By using a mixture of distributions, each with its own tangent space, we are able to ensure the consistency of the model while avoiding most of the linearization error caused by a single tangent space. The approach has

Table 8 Generalization of velocity estimation results

Method	Subject 1		Subject 5		Subject 6		Subject 7		Subject 8		Subject 9		Subject 11		Mean	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
GD	5.433	5.435	5.453	5.451	5.452	5.451	5.509	5.502	5.408	5.413	5.442	5.442	5.410	5.415	5.444	5.444
LGD	6.409	6.410	6.421	6.421	6.453	6.452	6.516	6.511	6.375	6.380	6.443	6.444	6.409	6.411	6.432	6.433
Ours block	9.360	11.925	10.170	7.449	10.503	5.039	11.319	2.129	8.943	12.188	9.854	9.597	9.495	11.793	9.949	8.589
Ours OAS	12.458	14.099	13.020	10.808	13.718	8.889	14.236	5.740	12.245	17.211	12.624	12.599	12.439	15.485	12.963	12.119

Log-likelihood evaluation of $p(\log_{x_t}(x_{t+1})|\mathcal{V}^*)$ on a single subject for testing and the rest for training. The subject indicated in the table is the subject used for testing. We compare a Gaussian diffusion (GD) strategy, a local Gaussian diffusion (LGD) against both our approaches. For fairness we only consider the $\mathcal{V}_1$ manifold, as Gaussian diffusion approaches do not use additional velocity information

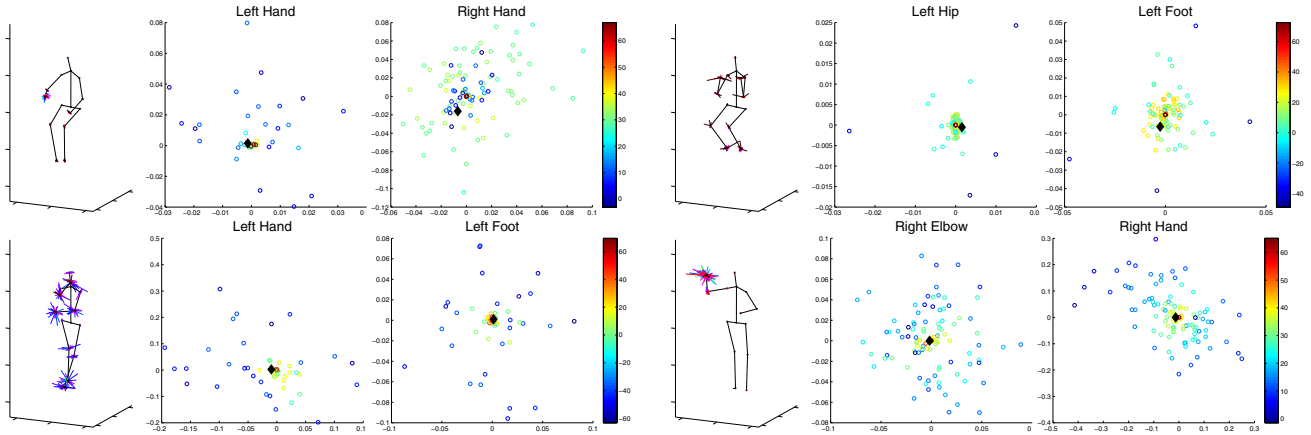


Fig. 9 Qualitative examples. Several examples from the test set using the 15 % subsampled model on the $\mathcal{V}_1$ manifold. We visualize the ground truth and 100 samples from our model in 3D. For visualization purposes the velocity is scaled by a factor $10\times$ and the samples are scaled by $3\times$. We also show the distribution of the samples on the tangent space for some of the joints, scored by their log-likelihood with the ground truth as a *black diamond*

been experimentally validated on various synthetic examples that highlight their ability to both correctly approximate manifold distributions and discover the underlying data structure. Furthermore, the approach has been tested on a large and complex dataset, where it is shown to outperform the traditionally used Euclidean Gaussian Mixture Model, von Mises–Fisher distributions and an approach using a single tangent space.

As a particular example, we have deeply studied the use of the model as a 3D pose tracking prior, and have shown it greatly outperforms the standard Gaussian diffusion prior. Additionally, by using shrinkage covariance estimation algorithms we are able to gain both robustness to poor data, and use more expressive covariance matrices.

Future works include exploiting the proposed algorithm on different manifolds and datasets. We have presented results using Gaussian distributions and have focused on the S^2 manifold. However, the algorithm presented here should work with any distribution and on any manifold for which the exponential and logarithmic map operators are provided, as shown on a quadratic surface. For example, it could be possible to initially estimate unknown and non-parameterizable manifolds Brand (2003), and use approximate operators Freifeld and Black (2012).

Acknowledgements We would like to thank the three anonymous reviewers for their insights and comments that have significantly contributed to improving this manuscript. This work was partly funded by the Spanish MINECO project RobInstruct TIN2014-58178-R and by the ERA-net CHISTERA project I-DRESS PCIN-2015-147.

Appendix 1: Derivation of Mixture Models on Riemannian Manifolds

We follow the standard expectation-maximization approach to maximize the log-likelihood of our model adapting it to Riemannian manifolds. For simplicity, we will not consider the Minimum Message Length criteria for model selection. We start out by defining the log-likelihood of the model $\lambda(x, \theta)$ and bounding it by Jensen's equality:

$$\begin{aligned}\lambda(x,\theta) &= \sum_{i=1}^{N} \log \sum_{k=1}^{K} \alpha_k \, p(x^{(i)}|\theta_k) \\ &\geq \sum_{i=1}^{N}\sum_{k=1}^{K} w_k^{(i)} \log \frac{\alpha_k \, p(x^{(i)}|\theta_k)}{w_k^{(i)}} = B(x,\theta)\ . \end{aligned} \tag{37}$$

with $w_k^{(i)}$ as auxiliary variables that represent membership probabilities. We can maximize over the lower bound $B(x, \theta)$ instead of the untractable full likelihood.

E-step

The E-step consists of maximizing the auxiliary terms $w_k^{(i)}$ which are the membership probabilities of the samples. This is done by solving:

$$\begin{aligned} \arg\max_{w} \quad & B(x,\theta) \\ \text{subject to} \quad & \sum_{i=1}^{N} w_k^{(i)} = 1,\ k = 1,\ldots,K \\ & w_k^{(i)} \geq 0,\ i = 1,\ldots,N,\ k = 1,\ldots,K\ . \end{aligned} \tag{38}$$

This is straight forward to do by computing the derivative and equating it to 0 to obtain the update rule for step t:

$$w_k^{(i)}(t) = \frac{\alpha_k(t-1)\, p(x_i|\theta_k(t-1))}{\sum_{k=1}^{K} \alpha_k(t-1)\, p(x_i|\theta_k(t-1))}\ . \tag{39}$$

M-step

In this step, we have fixed w and are updating the other parameters $\theta = (\mu, \Sigma)$ and α by solving:

$$\begin{aligned} \arg\max_{\theta,\alpha} \quad & B(x,\theta) \\ \text{subject to} \quad & \sum_{i=1}^{K} \alpha_k = 1 \\ & \alpha_k \geq 0, k = 1, \ldots, K \ . \end{aligned} \tag{40}$$

We shall follow the same approach as in the E-step and compute the partial derivatives to obtain the update rules. In particular, both α and Σ are straight forward to compute, and do not significantly deviate from the standard formulation. Thus for α we obtain:

$$\alpha_k(t) = \frac{1}{N} \sum_{i}^{N} w_k^{(i)} = \frac{w_k}{N} \ , \tag{41}$$

and for Σ:

$$\frac{\partial B(x,\theta)}{\partial \Sigma_k} = \frac{1}{2} \sum_{i=1}^{N} w_k^{(i)} \left(\log_{\mu_k}(x^{(i)}) \log_{\mu_k}(x^{(i)})^\top - \Sigma_k \right) \ , \tag{42}$$

and thus,

$$\Sigma_k(t) = \frac{\sum_{i=1}^{N} w_k^{(i)} \log_{\mu_k}(x^{(i)}) \log_{\mu_k}(x^{(i)})^\top}{\sum_{i=1}^{N} w_k^{(i)}} \ . \tag{43}$$

For the mean μ_k we can follow the same approach, however, due to the logarithmic map, it is slightly different to resolve. We start out by computing the partial derivative:

$$\frac{\partial B(x,\theta)}{\partial \mu_k} = \sum_{i=1}^{N} w_k^{(i)} \Sigma_k^{-1} \log_{\mu_k}(x^{(i)}) \frac{\partial \log_{\mu_k}(x^{(i)})}{\partial \mu_k} \ . \tag{44}$$

In general, there is no analytic solution to $\frac{\partial \log_{\mu_k}(x^{(i)})}{\partial \mu_k}$. However, under the assumption that $\frac{\partial \log_{\mu_k}(x^{(i)})}{\partial \mu_k} = c$ where $c \neq 0$ is a constant, and equating the partial derivative to 0, we can obtain:

$$\sum_{i=1}^{N} w_k^{(i)} \log_{\mu_k}(x^{(i)}) = 0 \ . \tag{45}$$

For simply connected and complete manifolds whose curvature is non-positive (i.e., Hadamard manifolds) and bounded from below, there exists one and only one Riemannian center of mass which is characterized by $E[\log_\mu(x)] = 0$ Darling (1996). Note that a compact and simply connected manifold with a non-positive and bounded from below curvature has no cut locus. In this case, as Eq. (45) is the discrete expectation of the weighted sum, we can establish the update rule for the mean by:

$$\mu_k(t) = \arg\min_{p} \sum_{i=1}^{N} d\left(\frac{N}{w_k} w_k^{(i)} x^{(i)}, \ p \right)^2 . \tag{46}$$

Note that this does not hold for the case in which there is a cut locus, in which case there may not be only one Riemannian center of mass. However, in practice, this approach will generally converge to the center of mass. We will use Eq. (46) in all cases.

Finally, we perform a numerical analysis of the error for the S^2 sphere by numerically computing $\frac{\partial \log_{\mu_k}(x^{(i)})}{\partial \mu_k}$ and visualizing the results. In particular we visualize the change of Frobenius norm of the Jacobian in Fig. 10. We can see that points near the origin have very little change in the derivative. Again, the use of multiple tangent planes favors configurations in which the points are close to the center, and thus, keeps the error produced by approximating the partial derivative to a constant within reasonable bounds.

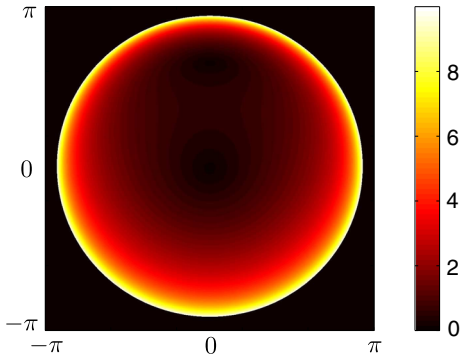


Fig. 10 Plot of the change Frobenius norm of $\frac{\partial \log_{\mu_k}(x^{(i)})}{\partial \mu_k}$. We compute the derivative numerically using a first order approximation as there is no analytic form. We can see for points near the center there is small change in the derivative and thus little error in the approximation we make by considering the derivative to be constant. For visualization purposes we only display points with a change of under 10 units

Appendix 2: Clustering with Von Mises–Fisher Distributions

Given a random vector x on the unit hypersphere of dimension $q - 1$, the probability density function of a von Mises–Fisher distribution with mean direction μ and concentration κ can be written as:

$$f_q(x|\mu,\kappa) = c_q(\kappa)e^{\kappa\mu^{\mathrm{T}}x} \quad ,$$
$$c_q(\kappa) = \frac{\kappa^{q/2-1}}{(2\pi)^{q/2}I_{q/2-1}(\kappa)} \quad , \tag{47}$$

where $\|\mu\| = 1$, and $I_{q/2-1}$ is the modified Bessel function of first kind and order $q/2-1$. Note that the concentration parameter κ is a single scalar that represents a uniform distribution on the sphere for $\kappa = 0$ and is unimodal for $\kappa > 0$.

The algorithm from Figueiredo and Jain (2002) can be modified to use von Mises–Fisher distributions by adapting the way the distributions are recalculated in the M-step. This can be computed by:

$$r_k = \sum_{i=1}^{N} w_k^{(i)} x^{(i)} \quad , \tag{48}$$
$$\mu_k(t) = \frac{r_k}{\|r_k\|} \quad , \tag{49}$$
$$\kappa_k(t) = \frac{\|r_k\|(q-\|r_k\|^2)}{1-\|r_k\|^2} \quad . \tag{50}$$

As there exists no analytic form of $I_{q/2}(\kappa_k(t))/I_{q/2-1}(\kappa_k(t)) = r_k$, the computation of $\kappa_k(t)$ is indeed an approximation Banerjee et al. (2005).

References

Andriluka, M., Roth, S., & Schiele, B. (2010). Monocular 3D pose estimation and tracking by detection. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Archambeau, C., & Verleysen, M. (2005). Manifold constrained finite gaussian mixtures. In: *Computational Intelligence and Bioinspired Systems* (pp. 820–828). Berlin: Springer.

Banerjee, A., Dhillon, I. S., Ghosh, J., Sra, S., & Ridgeway, G. (2005). Clustering on the unit hypersphere using von Mises–Fisher distributions. *Journal of Machine Learning Research*, *6*(9), 1345–1382.

Boothby, W. M. (2003). *An introduction to differentiable manifolds and riemannian geometry* (2nd ed.). New York: Academic Press.

Brand, M. (2003). Charting a manifold. In: *Neural Information Processing Systems* (pp. 961–968).

Brubaker, M. A., Salzmann, M., & Urtasun, R. (2012). A family of MCMC methods on implicitly defined manifolds. *Journal of Machine Learning Research*, *22*, 161–172.

do Carmo, M. P. (1992). *Riemannian geometry*. Boston: Birkhäuser.

Caseiro, R., Martins, P., Henriques, J. F., & Batista, J. (2012). A nonparametric riemannian framework on tensor field with application to foreground segmentation. *Pattern Recognition*, *45*(11), 3997–4017.

Caseiro, R., Martins, P., Henriques, J. F., Leite, F. S., & Batista, J. (2013). Rolling riemannian manifolds to solve the multi-class classification problem. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Chang, J., & Fisher III, J. W. (2013). Parallel sampling of dp mixture models using sub-cluster splits. In: *Neural Information Processing Systems* (pp. 620–628).

Chen, Y., Wiesel, A., Eldar, Y., & Hero, A. (2010). Shrinkage algorithms for mmse covariance estimation. *IEEE Transactions on Signal Processing*, *58*(10), 5016–5029.

Darling, R. (1996). Martingales on noncompact manifolds: Maximal inequalities and prescribed limits. *Annales de l'IHP Probabilités et statistiques*, *32*(4), 431–454.

Davis, B. C., Bullitt, E., Fletcher, P. T., & Joshi, S. (2007). Population shape regression from random design data. In: *International Conference on Computer Vision*.

Dedieu, J. P., & Nowicki, D. (2005). Symplectic methods for the approximation of the exponential map and the newton iteration on riemannian submanifolds. *Journal of Complexity*, *21*(4), 487–501.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1–38.

Deutscher, J., & Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal of Computer Vision*, *61*(2), 185–205.

Figueiredo, M., & Jain, A. (2002). Unsupervised learning of finite mixture models. *IEEE Transactions Pattern Analylis and Machine Intelligence*, *24*(3), 381–396.

Fletcher, P., Lu, C., Pizer, S., & Joshi, S. (2004). Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, *23*(8), 995–1005.

Freifeld, O., & Black, M. J. (2012). Lie bodies: A manifold representation of 3D human shape. In: *European Conference on Computer Vision*.

Gall, J., Rosenhahn, B., Brox, T., & Seidel, H. P. (2010). Optimization and filtering for human motion capture. *International Journal of Computer Vision*, *87*, 75–92.

Harandi, M., Sanderson, C., Hartley, R., & Lovell, B. (2012). Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In: *European Conference on Computer Vision*.

Harandi, M. T., Salzmann, M., & Hartley, R. (2014). From manifold to manifold: Geometry-aware dimensionality reduction for spd matrices. In: *European Conference on Computer Vision*.

Hauberg, S., Sommer, S., & Pedersen, K. S. (2012). Natural metrics and least-committed priors for articulated tracking. *Image and Vision Computing*, *30*(6), 453–461.

Huckemann, S., Hotz, T., & Munk, A. (2010). Intrinsic shape analysis: Geodesic PCA for Riemannian manifolds modulo isometric lie group actions. *Statistica Sinica*, *20*, 1–100.

Ionescu, C., Li, F., & Sminchisescu, C. (2011). Latent structured models for human pose estimation. In: *International Conference on Computer Vision*.

Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions Pattern Analylis and Machine Intelligence*, *36*(7), 1325–1339.

Jain, S., & Govindu, V. (2013). Efficient higher-order clustering on the grassmann manifold. In: *International Conference on Computer Vision*.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., & Harandi, M. (2013). Kernel methods on the riemannian manifold of symmetric positive definite matrices. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Jayasumana, S., Hartley, R., Salzmann, M., Li, H., & Harandi, M. (2015). Kernel methods on Riemannian manifolds with Gaussian RBF Kernels. In: *IEEE Transactions Pattern Analylis and Machine Intelligence*.

Karcher, H. (1977). Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, *30*(5), 509–541.

Lawrence, N. D. (2005). Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, *6*, 1783–1816.

Lawrence, N. D., & Moore, A. J. (2007). Hierarchical Gaussian process latent variable models. In: *International Conference in Machine Learning*.

Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, *88*(2), 365–411.

Ledoit, O., & Wolf, M. (2011). Nonlinear shrinkage estimation of large-dimensional covariance matrices. Institute for Empirical Research in Economics University of Zurich Working Paper (515).

Lenglet, C., Rousson, M., Deriche, R., & Faugeras, O. (2006). Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*, *25*(3), 423–444.

Li, R., Tian, T. P., Sclaroff, S., & Yang, M. H. (2010). 3D human motion tracking with a coordinated mixture of factor analyzers. *International Journal of Computer Vision*, *87*(1–2), 170–190.

Moeslund, T. B., & Granum, E. (2001). A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, *81*(3), 231–268.

Moeslund, T. B., Hilton, A., & Krüger, V. (2006). A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, *104*, 90–126.

Muralidharan, P., & Fletcher, P. T. (2012). Sasaki metrics for analysis of longitudinal data on manifolds. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Ozakin, A., & Gray, A. (2009). Submanifold density estimation. In: *Neural Information Processing Systems* (pp. 1375–1382).

Pelletier, B. (2005). Kernel density estimation on Riemannian manifolds. *Statistics & Probability Letters*, *73*(3), 297–304.

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

Pennec, X. (2009). Statistical computing on manifolds: From riemannian geometry to computational anatomy. In: *Emerging Trends in Visual Computing* (pp. 347–386). Berlin: Springer.

Pennec, X., Fillard, P., & Ayache, N. (2006). A Riemannian framework for tensor computing. *International Journal of Computer Vision*, *66*(1), 41–66.

Quiñonero-candela, J., Rasmussen, C. E., & Herbrich, R. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, *6*, 1939–1959.

Said, S., Courtry, N., Bihan, N. L., & Sangwine, S. (2007). Exact principal geodesic analysis for data on $SO(3)$. In: *European Signal Processing Conference*.

Sanin, A., Sanderson, C., Harandi, M., & Lovell, B. (2012). K-tangent spaces on riemannian manifolds for improved pedestrian detection. In: *International Conference on Image Processing*.

Sasaki, S. (1958). On the differential geometry of tangent bundles of riewannian manifolds. *Tohoku Mathematical Journal, Second Series*, *10*(3), 338–354.

Schäfer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, *4*(1), 32.

Shirazi, S., Harandi, M., Sanderson, C., Alavi, A., & Lovell, B. (2012). Clustering on grassmann manifolds via kernel embedding with application to action analysis. In: *International Conference on Image Processing*.

Sigal, L., Bhatia, S., Roth, S., Black, M., & Isard, M. (2004). Tracking loose-limbed people. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Sigal, L., Isard, M., Haussecker, H. W., & Black, M. J. (2012). Loose-limbed people: Estimating 3D human pose and motion using non-parametric belief propagation. *International Journal of Computer Vision*, *98*(1), 15–48.

Simo-Serra, E., Quattoni, A., Torras, C., & Moreno-Noguer, F. (2013). A joint model for 2D and 3D pose estimation from a single image. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Simo-Serra, E., Ramisa, A., Alenyà, G., Torras, C., & Moreno-Noguer, F. (2012). Single image 3D human pose estimation from noisy observations. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Simo-Serra, E., Torras, C., & Moreno-Noguer, F. (2014). Geodesic finite mixture models. In: *British Machine Vision Conference*.

Simo-Serra, E., Torras, C., & Moreno-Noguer, F. (2015). Lie algebra-based kinematic prior for 3D human pose tracking. In: *International Conference on Machine Vision Applications*.

Sivalingam, R., Boley, D., Morellas, V., & Papanikolopoulos, N. (2010). Tensor sparse coding for region covariances. In: *European Conference on Computer Vision*.

Sminchisescu, C., & Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, *22*(6), 371–391. Special issue on Visual Analysis of Human Movement.

Sommer, S. (2015). Anisotropic distributions on manifolds: Template estimation and most probable paths. In: *Information Processing in Medical Imaging*. Lecture Notes in Computer Science. Berlin: Springer.

Sommer, S., Lauze, F., Hauberg, S., & Nielsen, M. (2010). Manifold valued statistics, exact principal geodesic analysis and the effect of linear approximations. In: *European Conference on Computer Vision*.

Sommer, S., Lauze, F., & Nielsen, M. (2014). Optimization over geodesics for exact principal geodesic analysis. *Advances in Computational Mathematics*, *40*(2), 283–313.

Straub, J., Chang, J., Freifeld, O., & Fisher III, J. W. (2015). A dirichlet process mixture model for spherical data. In: *International Conference on Artificial Intelligence and Statistics*.

Taylor, G., Sigal, L., Fleet, D., & Hinton, G. (2010). Dynamical binary latent variable models for 3d human pose tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Tosato, D., Farenzena, M., Cristani, M., Spera, M., & Murino, V. (2010). Multi-class classification on riemannian manifolds for video surveillance. In: *European Conference on Computer Vision* (pp. 378–391).

Tosato, D., Spera, M., Cristani, M., & Murino, V. (2013). Characterizing humans on riemannian manifolds. *IEEE Transactions Pattern Analylis and Machine Intelligence*, *35*(8), 1972–1984.

Tournier, M., Wu, X., Courty, N., Arnaud, E., & Reveret, L. (2009). Motion compression using principal geodesics analysis. *Computer Graphics Forum*, *28*(2), 355–364.

Turaga, P., Veeraraghavan, A., Srivastava, A., & Chellappa, R. (2011). Statistical computations on grassmann and stiefel manifolds for image and video-based recognition. *IEEE Transactions Pattern Analylis and Machine Intelligence*, *33*(11), 2273–2286.

Tuzel, O., Porikli, F., & Meer, P. (2008). Pedestrian detection via classification on Riemannian manifolds. *IEEE Transactions Pattern Analylis and Machine Intelligence*, *30*(10), 1713–1727.

Urtasun, R., Fleet, D. J., & Fua, P. (2006). 3D people tracking with gaussian process dynamical models. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Urtasun, R., Fleet, D. J., & Lawrence, N. D. (2007). Modeling human locomotion with topologically constrained latent variable models. In: *Proceedings of the 2nd Conference on Human Motion: Understanding, Modeling, Capture and Animation*.

Varol, A., Salzmann, M., Fua, P., & Urtasun, R. (2012). A constrained latent variable model. In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Wallace, C. S., & Freeman, P. R. (1987). Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, 240–265.

Wang, J., Fleet, D., & Hertzmann, A. (2005). Gaussian process dynamical models. In: *Neural Information Processing Systems*.

Yao, A., Gall, J., Gool, L. V., & Urtasun, R. (2011). Learning probabilistic non-linear latent variable models for tracking complex activities. In: *Neural Information Processing Systems*.

Zhang, M., & Fletcher, P. T. (2013). Probabilistic principal geodesic analysis. In: *Neural Information Processing Systems* (pp. 1178–1186).