Noname manuscript No.
(will be inserted by the editor)

Unscented Kalman Filtering on Riemannian Manifolds

Søren Hauberg · François Lauze · Kim Steenstrup Pedersen

the date of receipt and acceptance should be inserted later

Abstract In recent years there has been a growing interest in problems, where either the observed data or hidden state variables are confined to a known Riemannian manifold. In sequential data analysis this interest has also been growing, but rather crude algorithms have been applied: either Monte Carlo filters or brute-force discretisations. These approaches scale poorly and clearly show a missing gap: no generic analogues to Kalman filters are currently available in non-Euclidean domains. In this paper, we remedy this issue by first generalising the unscented transform and then the unscented Kalman filter to Riemannian manifolds. As the Kalman filter can be viewed as an optimisation algorithm akin to the Gauss-Newton method, our algorithm also provides a general-purpose optimisation framework on manifolds. We illustrate the suggested method on synthetic data to study robustness and convergence, on a region tracking problem using covariance features, an articulated tracking problem, a mean value optimisation and a pose optimisation problem.

Keywords Riemannian Manifolds · Unscented Kalman Filter · Filtering Theory · Optimisation on Manifolds

S. Hauberg
Perceiving Systems, Max Planck Institute for Intelligent Systems
E-mail: soren.hauberg@tue.mpg.de

F. Lauze
Dept. of Computer Science, University of Copenhagen
E-mail: francois@diku.dk

K.S. Pedersen
Dept. of Computer Science, University of Copenhagen
E-mail: kimstp@diku.dk

1 Modelling with Manifolds

In many statistical problems it is becoming increasingly common to model non-linearities by confining parts of the model to a Riemannian manifold. This often provides better and more natural metrics, which has direct impact on the statistical models. The benefits of having good metrics have led manifolds to be used in a wide variety of models. Sometimes the observed data itself lives on a manifold, e.g. in Diffusion Tensor Imaging [9, 31] and shape analysis [10, 20]. Other times the hidden state variables of a generative model are confined to a non-Euclidean domain, e.g. in image segmentation [6] and human motion modelling [12, 15].

One notable downside to working with manifolds is the lack of many basic tools known from the Euclidean domain. While generalisations of mean values and covariances [30], as well as principal component analysis [10, 40] are available, most remaining tools are still missing. In this paper we tackle one of the most fundamental models for sequential data analysis: the Kalman filter. Our approach is based on the unscented Kalman filter (UKF) [17], which is a widely applied generalisation of the linear Kalman filter. This turns out to be a perfect fit for Riemannian manifolds as the unscented transform is readily generalisable. Furthermore, our approach is fairly easy to implement, unlike many other algorithms working on manifolds.

This paper is structured as follows. In the next section we discuss related work on filtering on manifolds and thereafter we provide a brief introduction to Riemannian manifolds including basic statistics (sec. 2). We present the theoretical contribution of the paper by presenting a generalisation of the unscented transform and the unscented Kalman filter for Riemannian manifolds (sec. 3). To illustrate the applicability of the new filter, we develop a series of demonstrations in sec. 4. The paper is concluded in sec. 5 with a discussion of further developments. Appendix A and B contain background material on Riemannian geometry and numerical algorithms for computing exponential maps and parallel transports. Finally, appendix C contains the proof of proposition 1.

1.1 Filtering on Manifolds

Filtering is the task of estimating the moments of the hidden state variable of a non-linear dynamical system [4],

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) \equiv \hat{f}(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) \ , \tag{1}$$

$$\mathbf{y}_t = h(\mathbf{x}_t) \quad \equiv \hat{h}(\mathbf{x}_t, \mathbf{n}_t) \ , \tag{2}$$

where $\mathbf{x}_t$ is the hidden state and $\mathbf{y}_t$ is the observation. The *process noise* $\mathbf{v}_t$ and *observation noise* $\mathbf{n}_t$ determine the stochastic nature of the system. To ease notation, we shall omit the noise terms in the rest of the paper. The (often non-linear) functions f and h respectively determine the system dynamics and relate the state to the observation. In the Euclidean case, this problem can be solved in closed-form when $\mathbf{x}_t$ is discrete using hidden Markov models [34], and by the Kalman filter [18] when the noise is additive and Gaussian with f and h linear. For other models, approximation schemes, such as particle filters [4] or extended [25] and unscented Kalman filters [17], are required.

When $\mathbf{x}_t$ is confined to a Riemannian manifold $\mathcal{M}$ the scenario is more difficult due to the inherent non-linearities of the state space. Srivastava and Klassen [41] note that "the classical Kalman-filtering framework does not apply" on Riemannian manifolds, so they apply a particle filter for estimating time-varying subspaces on the Grassmann manifold. Others have arrived at similar conclusions on other manifolds [15, 24, 26, 48] as this filter generalises easily. On the downside, the filter is stochastic, which makes it hard to analyse. Another option, advocated by Tidefelt and Schön [43], is to discretise $\mathcal{M}$ and use an ordinary hidden Markov model on the discrete domain. Both approaches are affected with the curse of dimensionality and the complexity scales exponentially with the dimension of the state space; the former approach requires more particles and the latter more discretisation bins as the dimension increases. For many problems the computational burden of these approaches becomes too much and alternatives are needed.

One such approach is the *mean shift*, which has been generalised to Riemannian spaces [42]. This does not solve the filtering problem as defined above, yet it does provide an efficient solution to many tracking problems.

In Euclidean domains, the Kalman filter provides an efficient and robust solution to the filtering problem that scales well, when the observation has a unimodal distribution. In general, this filter is, however, not applicable to Riemannian manifolds, though some work has been done on selected Lie groups [22, 23, 45]. Tyagi and Davis [45] have shown how this filter can be applied for a specific dynamical model on the Lie group of positive definite symmetric matrices. Similarly, Kraft [22] and Kwon et al. [23] show how to apply the unscented Kalman filter on the Lie groups of unit Quaternions and $SO(3)$ and $SE(3)$, respectively. Sipos [38] extends the work of Kraft by showing how to implement the computationally more efficient "square root filter" on the Lie groups of unit Quaternions. While these approaches have some similarity to our work, they are based on specific knowledge of the Lie groups and cannot easily be generalised to other domains. Examples of such domains include the *bicycle chain* shape model [39] and the *kinematic manifold* for human poses [15, 16].

In this paper we provide a filter more general than the ones suggested for Lie groups, as it works for at least any geodesically complete Riemannian manifolds. First, we pause to review some basic tools from Riemannian geometry as these are needed to fully grasp the details of the filter. It should, however, be noted that practical implementation of the filter only requires knowledge of a few basic operations on the manifold.

2 Basic Tools on Riemannian Manifolds

In this section we recall some of the elementary aspects of Riemannian geometry. More details can be found in appendix A. We closely follow [5] and [30]. Riemannian geometry studies smooth manifolds endowed with a Riemannian metric. A metric on a manifold M is a smoothly varying inner product given in the tangent space $T_{\mathbf{x}}\mathcal{M}$ at each point $\mathbf{x}$ on the manifold. The tangent space at a point $\mathbf{x}$ of $\mathcal{M}$ is a Euclidean space, which locally approximates the manifold. For this reason, the inner product provides an infinitesimal metric, denoted $\langle -, - \rangle_{\mathbf{x}}$, and associated norm $\| - \|_{\mathbf{x}}$ which can be integrated along the manifold. Thus, the length L of a curve $\alpha : [0, 1] \to \mathcal{M}$ connecting two points $\mathbf{x}$ and $\mathbf{y}$ on $\mathcal{M}$ ($\alpha(0) = \mathbf{x}$, $\alpha(1) = \mathbf{y}$) is defined by integrating the size of the curve derivative with respect to the local metric,

$$L(\alpha) = \int_0^1 \|\alpha'(\tau)\| \mathrm{d}\tau = \int_0^1 \langle \alpha'(\tau), \alpha'(\tau) \rangle_{\alpha(\tau)}^{\frac{1}{2}} \mathrm{d}\tau \ , \tag{3}$$

where $\alpha' \in T_\alpha\mathcal{M}$ denotes the curve derivative. The distance $d(\mathbf{x},\mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ is the infimum

$$d(\mathbf{x},\mathbf{y}) = \inf_\alpha L(\alpha), \quad \alpha(0)=\mathbf{x}, \quad \alpha(1)=\mathbf{y}. \tag{4}$$

Next, *geodesics* are critical points of the curve energy

$$E(\alpha) = \frac{1}{2}\int_0^1 \|\alpha'(\tau)\|^2 \mathrm{d}\tau \tag{5}$$

and they also optimise the length functional eq. 3.

Many operations are defined in the Euclidean tangent space $T_\mathbf{x}\mathcal{M}$ and there are mappings back and forth between the manifold and the tangent space. The Riemannian *exponential map* at a point $\mathbf{x}\in\mathcal{M}$ maps a tangent vector $\mathbf{v}\in T_\mathbf{x}\mathcal{M}$ to the point $\mathbf{y}=\mathrm{Exp}_\mathbf{x}(\mathbf{v})\in\mathcal{M}$ such that the curve $t\mapsto \mathrm{Exp}_\mathbf{x}(t\mathbf{v})$ is a geodesic going from $\mathbf{x}$ to $\mathbf{y}$ with length $\|\mathbf{v}\|$. It is in general only defined in a neighbourhood of the origin of $T_\mathbf{x}\mathcal{M}$. The inverse mapping, which maps $\mathbf{y}$ to $\mathbf{v}$, is the Riemannian *logarithm map*, denoted $\mathrm{Log}_\mathbf{x}(\mathbf{y})$. It is in general only defined in a neighbourhood of $\mathbf{x}$. $\mathrm{Exp}_\mathbf{x}$ is the straightest local parametrisation of $\mathcal{M}$ in a neighbourhood of $\mathbf{x}$ in the sense that it is the one that locally least deforms distances around $\mathbf{x}$. The logarithm and exponential maps are illustrated in fig. 1a.

$\mathcal{M}$ is called *geodesically complete* if the domain of definition of a geodesic can be extended to $\mathbb{R}$, i.e, if $\mathrm{Exp}_\mathbf{x}$ is defined on all of $T_\mathbf{x}\mathcal{M}$, for each $\mathbf{x}\in\mathcal{M}$. Then the Hopf-Rinow theorem asserts that such a manifold is complete for the above defined distance, eq. 4, and for each pair $(\mathbf{x},\mathbf{y})$ of points of $\mathcal{M}$ there exists at least one geodesic α joining $\mathbf{x}$ and $\mathbf{y}$ such that $L(\alpha)=d(\mathbf{x},\mathbf{y})$. For that reason we will assume that our manifold $\mathcal{M}$ is geodesically complete.

Generalising basic statistics to Riemannian manifolds is straightforward [30]. The empirical mean of a set of data points $\mathbf{x}=\{x_1\dots,x_K\}$ is defined as the point on the manifold that minimises the sum of squared distances:

$$\mathrm{E}[\mathbf{x}] = \arg\min_\mu \sum_{k=1}^K d^2(\mathbf{x}_k,\mu) \ . \tag{6}$$

Unlike the Euclidean case, such a mean is not necessarily unique; local optima of eq. 6 are known as *Karcher means*, while a global optimum is called the *Fréchet mean* [19]. When $\mu=\mathrm{E}[\mathbf{x}]$ exists, the empirical covariance is generalised as

$$\mathbf{P}_\mu = \frac{1}{K}\sum_{k=1}^K \mathrm{Log}_\mu(\mathbf{x}_k)\,\mathrm{Log}_\mu(\mathbf{x}_k)^T \ . \tag{7}$$

The transposition operation is taken with respect to an orthonormal basis of $T_\mu\mathcal{M}$. $\mathbf{P}_\mu$ is a bilinear symmetric (and positive semi-definite) operator on $T_\mu\mathcal{M}$ defined as follows: Given $v,w\in T_\mu\mathcal{M}$,

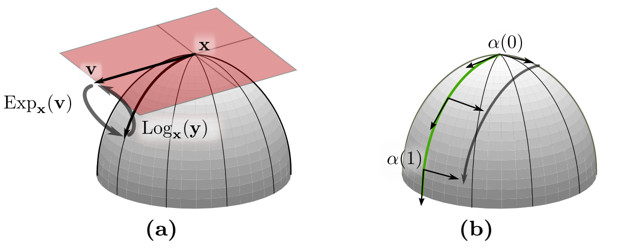


Fig. 1 Graphical illustration of basic manifold operations. (a) The exponential and logarithm maps. (b) The parallel transport for moving vectors along a curve.

$$\begin{aligned}\mathbf{P}_\mu(v,w) &= v^T\frac{1}{K}\sum_{k=1}^K \mathrm{Log}_\mu(\mathbf{x}_k)\,\mathrm{Log}_\mu(\mathbf{x}_k)^T\, w \qquad (8)\\ &= \frac{1}{K}\sum_{k=1}^K v^T\,\mathrm{Log}_\mu(\mathbf{x}_k)\,\mathrm{Log}_\mu(\mathbf{x}_k)^T\, w\\ &= \frac{1}{K}\sum_{k=1}^K \langle v, \mathrm{Log}_\mu\mathbf{x}_k\rangle_\mu\langle w,\mathrm{Log}_\mu\mathbf{x}_k\rangle_\mu .\end{aligned}$$

Given a curve $\alpha:[0,1]\to\mathcal{M}$ there exist *isometries* (thus preserving inner products) $P_t : T_{\alpha(0)}\mathcal{M}\to T_{\alpha(t)}\mathcal{M}$ called the *parallel transport along* α. Parallel transport extends naturally to more general objects than vectors, called tensors. The parallel transport is the straightest, or least deforming way to move geometric objects along curves. Coupled with the exponential map, it provides the straightest way to move a geometric object from one point of the manifold to a neighbouring point via the geodesic curve that joins them. This is usually defined via the *Levi-Civita connection* and associated *covariant derivatives* uniquely associated to the metric. They are defined in appendix A and the parallel transport is illustrated in fig. 1b.

A fundamental point for the construction of our filters is that parallel transport extends to multilinear operators, and in particular to bilinear symmetric ones and this allows us to *transport covariances along curves.* The following proposition provides a construction for this special case. The proof is given in appendix C.

Proposition 1 *Let $M_\mathbf{x}$ be a bilinear mapping on $T_\mathbf{x}\mathcal{M}$ and $\alpha:[0,1]\to\mathcal{M}$ a differentiable curve on $\mathcal{M}$ with $\alpha(0)=\mathbf{x}$. Since $M_\mathbf{x}$ is symmetric, it can be written as*

$$M_\mathbf{x} = \sum_{m=1}^M \lambda_m v_m v_m^T \ , \tag{9}$$

where $(v_1,\dots,v_M)$ is an orthonormal basis of $T_\mathbf{x}\mathcal{M}$ and the λ_ms are the eigenvalues of $M_\mathbf{x}$ associated to the eigenvectors v_m. Let $v_m(t)=P_t(v_m)$ be the parallel *transport of v_m along α ($m = 1, \ldots, M$). With this,*

$$M_t = \sum_{m=1}^{M} \lambda_m v_m(t) v_m(t)^T \tag{10}$$

is the parallel transport of $M_{\mathbf{x}}$ along α.

3 The Manifold UKF

We now have the preliminaries settled and are ready to design the unscented Kalman filter on Riemannian manifolds. We shall first generalise the unscented transform and then we provide the new filter.

3.1 The Unscented Transform

The *unscented transform* [17] is a method for estimating the mean and covariance of a distribution undergoing a non-linear transformation. Given a stochastic variable $\mathbf{x}$, the idea is to pick a set of *sigma points* that fully describe the mean and covariance of $\mathbf{x}$ and then let each sigma point undergo a non-linear transformation f. The mean and covariance of $f(\mathbf{x})$ can then be estimated by computing the sample mean and covariance of the transformed sigma points.

In more detail, let $\bar{\mathbf{x}}$ and $\mathbf{P}$ denote the mean and covariance of the M dimensional variable $\mathbf{x}$. The sigma points are then calculated as

$$\sigma^{(0)} = \bar{\mathbf{x}} \tag{11}$$

$$\sigma^{(m)} = \bar{\mathbf{x}} \pm \left(\sqrt{(M+\lambda)\mathbf{P}}\right)_m , \quad m = 1, \ldots, 2M , \tag{12}$$

where $\left(\sqrt{\cdot}\right)_m$ denotes the m^{th} column of the Cholesky decomposition and λ is a parameter for controlling the distance between the sigma points and the mean value. The sigma points are illustrated in fig. 2a. The mean and covariance of $f(\mathbf{x})$ can then be estimated as

$$\mathrm{E}[f(\mathbf{x})] \approx \mu = \sum_{m=0}^{2M} w_m f(\sigma^{(m)}) , \tag{13}$$

$$\mathrm{cov}[f(\mathbf{x})] \approx \sum_{m=0}^{2M} w_m (f(\sigma^{(m)}) - \mu)(f(\sigma^{(m)}) - \mu)^T, \tag{14}$$

where the weights are defined as

$$w_0 = \frac{\lambda}{\lambda + M} , \tag{15}$$

$$w_m = \frac{1}{2(\lambda + M)} , \quad m = 1, \ldots, 2M . \tag{16}$$

These equations are enough to approximate the correct mean to third order and covariance to the second order [17].

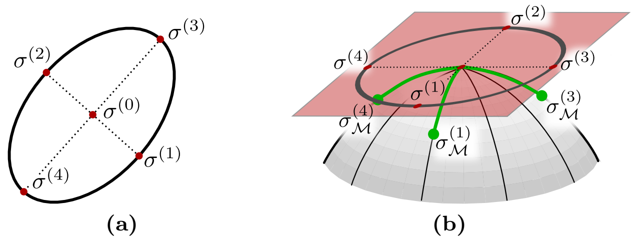


Fig. 2 An illustration of the sigma points. The ellipse represents a covariance. (a) Sigma points in the Euclidean case. (b) Sigma points in the tangent space and on the manifold.

3.1.1 Generalisations

We now consider two different approaches to generalising the unscented transform for Riemannian manifolds; both approaches are based on the same basic observation. Consider a stochastic variable $\mathbf{x} \in \mathcal{M}$ with mean value $\bar{\mathbf{x}}$ and covariance $\mathbf{P}$ expressed in the basis of the tangent space at $\bar{\mathbf{x}}$. Let $\sigma^{(0:2M)} = \{\sigma^{(0)}, \ldots, \sigma^{(2M)}\}$ denote the sigma points of the covariance $\mathbf{P}$ calculated in the (Euclidean) tangent space. Inspection of eq. 7 reveals that

$$\sigma_{\mathcal{M}}^{(m)} = \mathrm{Exp}_{\bar{\mathbf{x}}}(\sigma^{(m)}) , \quad m = 0, \ldots, 2M \tag{17}$$

captures both the mean and covariance. We, thus, have two sets of sigma points that capture the statistics; one set on the manifold and one in the tangent space. These are illustrated in fig. 2b. This gives rise to two different, but equally useful, unscented transforms that we shall both discuss.

First, we consider the case where the non-linear mapping, $f : \mathcal{M}_1 \to \mathcal{M}_2$, moves the sigma points from the manifold to a possibly different (possibly Euclidean) manifold. The mean value can then be estimated by computing the average of the transformed sigma points as discussed in sec. 2, i.e.

$$\begin{aligned} \mathrm{E}[f(\mathbf{x})] &\approx \mu_{\mathcal{M}_2} \\ &= \arg\min_{\mathbf{q} \in \mathcal{M}_2} \sum_{m=0}^{2M} w_m d^2(f(\sigma_{\mathcal{M}_1}^{(m)}), \mathbf{q}) , \end{aligned} \tag{18}$$

where $d(\cdot, \cdot)$ denotes geodesic distance on $\mathcal{M}_2$. The covariance can be estimated in the tangent space of $\mu_{\mathcal{M}_2}$ using eq. 7,

$$\begin{aligned} &\mathrm{cov}[f(\mathbf{x})] \\ &\approx \sum_{m=0}^{2M} w_m \mathrm{Log}_{\mu_{\mathcal{M}_2}}(f(\sigma_{\mathcal{M}_1}^{(m)}))\, \mathrm{Log}_{\mu_{\mathcal{M}_2}}(f(\sigma_{\mathcal{M}_1}^{(m)}))^T . \end{aligned} \tag{19}$$

A second generalisation considers the case where the non-linear mapping, $f : T_{\bar{\mathbf{x}}}\mathcal{M} \to T_{\bar{\mathbf{x}}}\mathcal{M}$, moves the sigma points in the tangent space. After this transformation a new mean and covariance can be calculated using ordinary Euclidean techniques in the tangent space, i.e.

$$\mathrm{E}[f(\mathbf{x})] \approx \mu_{T_{\bar{\mathbf{x}}}\mathcal{M}} = \sum_{m=0}^{2M} w_m f(\sigma^{(m)}) \ , \tag{20}$$

$$\mathrm{cov}[f(\mathbf{x})] \approx \sum_{m=0}^{2M} w_m (f(\sigma^{(m)}) - \mu_{T_{\bar{\mathbf{x}}}\mathcal{M}})(f(\sigma^{(m)}) - \mu_{T_{\bar{x}}\mathcal{M}})^T \ . \tag{21}$$

The mean value $\mu_{T_{\bar{\mathbf{x}}}\mathcal{M}}$ can readily be transferred back to the manifold as $\mu = \mathrm{Exp}_{\bar{\mathbf{x}}}(\mu_{T_{\bar{x}}\mathcal{M}})$; the covariance is transported using parallel transport defined in eq. 10, along the geodesic path $t \mapsto \mathrm{Exp}_{\bar{\mathbf{x}}}(t\mu_{T_{\bar{x}}\mathcal{M}})$, c.f. proposition 1.

3.2 The Unscented Kalman Filter

Before stating the Riemannian generalisations of the Kalman filter, we review the Euclidean techniques for optimal minimum mean-squared error (MMSE) filtering as it provides the basis for the generalisations.

Consider the dynamical system in eq. 1 and 2 with known initial mean and covariance

$$\bar{\mathbf{x}}_0 = \mathrm{E}[\mathbf{x}_0] \ , \tag{22}$$

$$\mathbf{P}_0 = \mathrm{cov}[\mathbf{x}_0] \ . \tag{23}$$

The Euclidean optimal minimum mean-squared error estimate of $\mathbf{x}_t$ can then be written as a linear interpolation between the prediction $\hat{\mathbf{x}}_t$ of $\mathbf{x}_t$ and the predicted observation $\hat{\mathbf{y}}_t$ [25],

$$\bar{\mathbf{x}}_t = \hat{\mathbf{x}}_t + \mathbf{K}(\mathbf{y}_t - \hat{\mathbf{y}}_t) \ , \tag{24}$$

where $\mathbf{K}$ is the so-called *Kalman gain*. Here the terms are calculated as

$$\hat{\mathbf{x}}_t = \mathrm{E}[f(\mathbf{x}_{t-1})] \ , \tag{25}$$

$$\mathbf{K} = \mathbf{P}_{\mathbf{xy}}\mathbf{P}_{\mathbf{yy}}^{-1} \ , \tag{26}$$

$$\hat{\mathbf{y}}_t = \mathrm{E}[h(\hat{\mathbf{x}}_t)] \ , \tag{27}$$

where $\mathbf{P}_{\mathbf{yy}}$ denotes the covariance of $\hat{\mathbf{y}}_t$ and $\mathbf{P}_{\mathbf{xy}}$ the cross-covariance of $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{y}}_t$. The covariance of the state estimate can also be propagated as

$$\mathbf{P}_t = \hat{\mathbf{P}}_t - \mathbf{K}\mathbf{P}_{\mathbf{yy}}\mathbf{K}^T \ , \tag{28}$$

where $\hat{\mathbf{P}}_t = \mathrm{cov}[f(\mathbf{x}_{t-1})]$. The above equations can, however, only be solved in closed-form when f and h are linear functions. One common approximation for the non-linear scenario is the unscented Kalman filter (UKF) [17].

3.2.1 UKF: The Euclidean Case

Let $\bar{\mathbf{x}}_{t-1}$ and $\mathbf{P}_{t-1}$ denote the mean and covariance of the state estimate at time $t-1$. A set of sigma points, $\sigma^{(0:2M)}$, can be calculated from these using the unscented transform, which allows us to estimate $\hat{\mathbf{x}}_t$

$$\hat{\mathbf{x}}_t \approx \sum_{m=0}^{2M} w_m f(\sigma^{(m)}) \ , \tag{29}$$

$$\hat{\mathbf{P}}_t \approx \sum_{m=0}^{2M} w_m (f(\sigma^{(m)}) - \hat{\mathbf{x}}_t)(f(\sigma^{(m)}) - \hat{\mathbf{x}}_t)^T \ . \tag{30}$$

Likewise, the unscented transform can be applied to estimate the effects of h:

$$\hat{\mathbf{y}}_t \approx \sum_{m=0}^{2M} w_m h(\sigma^{(m)}) \ . \tag{31}$$

The covariance and cross-covariance needed to compute the Kalman gain can also be readily approximated

$$\mathbf{P}_{\mathbf{yy}} \approx \sum_{m=0}^{2M} w_m (h(\sigma^{(m)}) - \hat{\mathbf{y}}_t)(h(\sigma^{(m)}) - \hat{\mathbf{y}}_t)^T \ , \tag{32}$$

$$\mathbf{P}_{\mathbf{xy}} \approx \sum_{m=0}^{2M} w_m (f(\sigma^{(m)}) - \hat{\mathbf{x}}_t)(h(\sigma^{(m)}) - \hat{\mathbf{y}}_t)^T \ . \tag{33}$$

These estimates are second-order accurate [17].

3.2.2 UKF: The Riemannian Case

We now use the same approach to generalise the Kalman filter to Riemannian state spaces. Here we shall assume that $h : \mathcal{M} \to \mathcal{M}_{\mathrm{obs}}$, where $\mathcal{M}_{\mathrm{obs}}$ denotes the observation space. The filter can then be expressed as the following steps, which will be elaborated later:

1. Use the Riemannian generalisation of the unscented transform to estimate the predicted state mean, $\hat{\mathbf{x}}_t = \mathrm{E}[f(\mathbf{x}_{t-1})]$, and covariance $\hat{\mathbf{P}}_t = \mathrm{cov}[f(\mathbf{x}_{t-1})]$.
2. Compute the Riemannian generalisation of the unscented transform of $\hat{\mathbf{P}}_t$ to estimate $\hat{\mathbf{y}}_t$, $\mathbf{P}_{\mathbf{yy}}$ and $\mathbf{P}_{\mathbf{xy}}$. Here, $\hat{\mathbf{y}}_t \in \mathcal{M}_{\mathrm{obs}}$, $\mathbf{P}_{\mathbf{yy}}$ describes covariance in $T_{\hat{\mathbf{y}}_t}\mathcal{M}_{\mathrm{obs}}$ and $\mathbf{P}_{\mathbf{xy}}$ describes the cross-covariance between the sigma points in $T_{\hat{\mathbf{x}}_t}\mathcal{M}$ and $T_{\hat{\mathbf{y}}_t}\mathcal{M}_{\mathrm{obs}}$.
3. Compute state updates $\bar{\mathbf{x}}_t$ and $\mathbf{P}_t$ according to eq. 24 and 28. These will be expressed in $T_{\hat{\mathbf{x}}_t}\mathcal{M}$.
4. Move $\bar{\mathbf{x}}_t$ to the manifold as $\mathrm{Exp}_{\hat{\mathbf{x}}_t}(\bar{\mathbf{x}}_t)$ and parallel transport $\mathbf{P}_t$ to the tangent space at this point according to proposition 1.

The above steps are essentially straight-forward generalisations of the Euclidean case. However, as two different generalisations of the unscented transform are available (one in the tangent space and one on the manifold), some details need further attention. In the next three sections, we discuss the first three steps in the above filter in greater detail.

3.2.3 Step 1: Dynamical Models

When the system is predicted according to the dynamical model, two general types of models are worth considering.

The first is when the dynamical model f moves the sigma points directly on the manifold, i.e. $f : \mathcal{M} \to \mathcal{M}$. In this case, each sigma point $\sigma_{\mathcal{M}}^{(n)}$ is propagated through f and a mean and covariance can be estimated according to eq. 18 and 19. This requires knowledge of the logarithm map on $\mathcal{M}$, but not of the parallel transport.

The second class of dynamical models worth considering is when the sigma points are moved in the tangent space, i.e. $f : T_{\bar{\mathbf{x}}_t}\mathcal{M} \to T_{\bar{\mathbf{x}}_t}\mathcal{M}$. In general, the dynamical model happens directly on the manifold and should not be expressed in the tangent space. However, if dynamics are simple (i.e. the identity function) or time-steps are small a first-order approximation in the tangent space can be convenient. When the dynamics are expressed in tangent space, the predicted mean and covariance can be estimated using ordinary Euclidean techniques. The mean value can be moved back to the manifold using the exponential map and then the covariance can be parallel transported to this mean value. This does not require knowledge of the logarithm map, but does require the parallel transport and exponential map.

3.2.4 Step 2: Observation Models

The unscented Kalman filter is a generative model: the h function must generate an observation for each input sigma point. It is, thus, reasonable to require that this function is given a valid state as input, i.e. the input should be confined to the state manifold $\mathcal{M}$. Hence, we will not consider the case where the input is in the tangent space of $\mathcal{M}$, giving $h : \mathcal{M} \to \mathcal{M}_{\text{obs}}$.[1]

Let $\sigma_{\mathcal{M}}^{(0:2M)}$ denote the sigma points corresponding to $\hat{\mathbf{P}}_t$. The mean $\hat{\mathbf{y}}_t$ of $h(\sigma_{\mathcal{M}}^{(0:2M)})$ is computed using eq. 18. The transformed sigma points are then lifted to the tangent space at $\hat{\mathbf{y}}_t$, and $\mathbf{P}_{\mathbf{yy}}$ and $\mathbf{P}_{\mathbf{xy}}$ are estimated as

$$\mathbf{P}_{\mathbf{yy}} \approx \sum_{m=0}^{2M} w_m \operatorname{Log}_{\hat{\mathbf{y}}_t}(h(\sigma_{\mathcal{M}}^{(m)})) \operatorname{Log}_{\hat{\mathbf{y}}_t}(h(\sigma_{\mathcal{M}}^{(m)}))^T, \quad (34)$$

$$\begin{aligned}\mathbf{P}_{\mathbf{xy}} &\approx \sum_{m=0}^{2M} w_m \operatorname{Log}_{\hat{\mathbf{x}}_t}(\sigma_{\mathcal{M}}^{(m)}) \operatorname{Log}_{\hat{\mathbf{y}}_t}(h(\sigma_{\mathcal{M}}^{(m)}))^T \\ &= \sum_{m=0}^{2M} w_m \sigma^{(m)} \operatorname{Log}_{\hat{\mathbf{y}}_t}(h(\sigma_{\mathcal{M}}^{(m)}))^T \ .\end{aligned} \quad (35)$$

When the observation manifold is $\mathbb{R}^N$, the logarithm map is no longer necessary and the above equations reduce to

$$\mathbf{P}_{\mathbf{yy}} \approx \sum_{m=0}^{2M} w_m (h(\sigma_{\mathcal{M}}^{(m)}) - \hat{\mathbf{y}}_t)(h(\sigma_{\mathcal{M}}^{(m)}) - \hat{\mathbf{y}}_t)^T \ , \quad (36)$$

$$\mathbf{P}_{\mathbf{xy}} \approx \sum_{m=0}^{2M} w_m \sigma^{(m)} (h(\sigma_{\mathcal{M}}^{(m)}) - \hat{\mathbf{y}}_t)^T \ . \quad (37)$$

When, moreover, the dynamics can be modelled in the tangent space of the current estimate $T_{\bar{\mathbf{x}}_t}\mathcal{M}$, no logarithm map is involved. This has great practical importance, as apart from relatively simple manifolds, computing the logarithm map generally requires solving an optimal control problem [5].

3.2.5 Step 3: State Update

Once covariances and cross-covariances have been computed in the two tangent spaces $T_{\hat{\mathbf{x}}_t}\mathcal{M}$ and $T_{\hat{\mathbf{y}}_t}\mathcal{M}_{\text{obs}}$, the Kalman gain can readily be computed using eq. 26. This provides a linear transformation between the two tangent spaces, so we need to express the state update equation (eq. 24) in the local coordinate systems of the tangent spaces. This is readily written as,

$$\bar{\mathbf{x}}_t = \hat{\mathbf{x}}_t + \mathbf{K} \operatorname{Log}_{\hat{\mathbf{y}}_t}(\mathbf{y}_t) \ . \quad (38)$$

The covariance of the state update can be computed using eq. 28.

4 Experimental Results

In this section, we provide a series of experiments to show that the filter is applicable in a wide range of scenarios and that it can be superior to the particle filter. First, we provide an experiment on synthetic data, where we vary the parameters of the data in order to study the sensitivity of the Riemannian UKF. Secondly, we build a region tracker based on region covariance features, which naturally lives on a Riemannian manifold [44]. Thirdly, we create an articulated tracker for estimating human poses. Due to constant bone lengths in

[1] In the rare cases when $h : T\mathcal{M} \to \mathcal{M}_{\text{obs}}$, we can consider $\hat{h}(\cdot) \equiv h(\operatorname{Exp}(\cdot))$ instead.

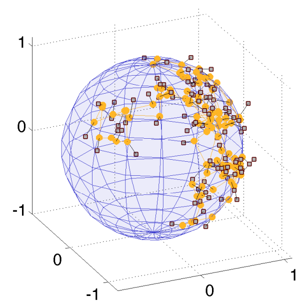


Fig. 3 Example of synthetic data. The hidden state is sampled according to a random walk on the sphere, while observations are sampled by adding Gaussian noise to the state. In this example, the state noise is $\gamma_{\text{state}} = 0.5/\sqrt{3}$ and the observation noise is $\gamma_{\text{obs}} = 0.1$.

the human body, our pose representation is naturally a Riemannian manifold [16]. Finally, we experiment with using the filter as an optimisation scheme, both on a synthetic mean value problem, and on a real-world pose fitting problem.

The source code for the first experiments are available as part of the supplementary material of the paper, while the articulated tracking code is available as part of the publicly available *HUMIM tracker* [14].

4.1 Synthetic Data

The first experiment we perform seeks to quantify how the suggested filter performs compared to a particle filter, when various parameters change. To keep the experiment under control, we use synthetic data, which we generate as follows. We let the state space $\mathcal{M}$ be an M dimensional sphere embedded in $\mathbb{R}^{M+1}$. We sample states following a first order Markov chain on the sphere by sampling an isotropic Gaussian in the tangent space, i.e.

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}) = \text{Exp}_{\mathbf{x}_{t-1}}\left(\mathcal{N}(\mathbf{0}, \gamma_{\text{state}}^2\mathbf{I})\right) \quad . \tag{39}$$

The observations are then sampled from a Gaussian distribution in the embedding space with mean value corresponding to the state, i.e.

$$\mathbf{y}_t \sim \mathcal{N}\left(\mathbf{x}_t, \gamma_{\text{obs}}^2\mathbf{I}\right) \quad . \tag{40}$$

An example of the synthetic data can be seen in fig. 3.

4.1.1 State Space Dimensionality

First, we fix the noise parameters for the dynamical and observational models and vary the dimensionality of the state space between 3 and 200. For each choice of dimensionality M, we run the Riemannian UKF and a particle filter with $2M+1$ particles and then with $10M$ particles. We choose $2M+1$ particles as this gives the same number of likelihood evaluations as in the UKF.

We measure the error of the estimated states as

$$\mathcal{E} = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{x}_t - \bar{\mathbf{x}}_t\| \quad , \tag{41}$$

where $\mathbf{x}_t$ and $\bar{\mathbf{x}}_t$ respectively denote the true and the estimated state at time t.

Fig. 4a shows the results of the experiment. As can be seen, the Riemannian UKF outperforms the particle filter, except for very low dimensional state spaces. This is not surprising as the particle filter is known to perform poorly in high dimensional state spaces.

4.1.2 Running Time

In the previous experiment we also recorded the running time of the different filters. These are reported in fig. 4b; as can be seen the Riemannian UKF scales much better than the particle filter, which again is not surprising.

4.1.3 Noise Sensitivity

Finally, we fix the state space dimensionality to $M = 30$ and vary the noise parameter in the dynamical model from $\gamma_{\text{state}} = \frac{0.001}{\sqrt{M}}$ to $\gamma_{\text{state}} = \frac{0.9}{\sqrt{M}}$. Again, we report in fig. 4c the error measure from eq. 41. As can be seen, the particle filter works best for small noise values, but is outperformed by the Riemannian UKF for larger noise values.

4.2 Tracking with Covariance Features

The *region covariance descriptor* [44] is a popular image patch descriptor, which computes a set of feature responses, e.g. *edges* and *colours*, for each pixel inside the region and then uses the covariance matrix of the features as a patch descriptor. As covariance matrices are positive definite they naturally have a manifold structure [44]; we denote this manifold Sym_+. This structure is simple enough to have closed-form exponential map given by

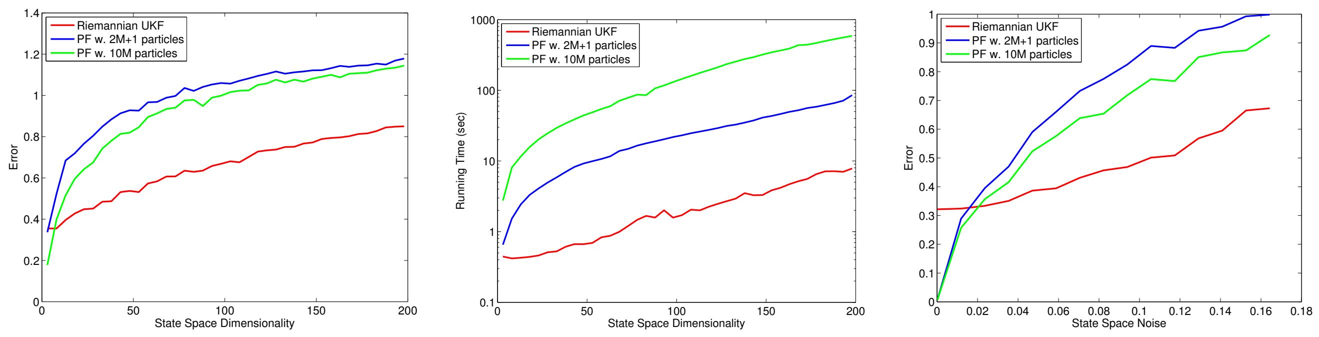
$$\text{Exp}_{\mathbf{x}}(\mathbf{v}) = \mathbf{x}\,\text{Expm}(\mathbf{x}^{-1}\mathbf{v}) \quad , \tag{42}$$

where $\text{Expm}(\cdot)$ is the matrix exponential

$$\text{Expm}(\mathbf{v}) = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{v}^k \quad . \tag{43}$$

Likewise, the logarithm map is given by

$$\text{Log}_{\mathbf{x}}(\mathbf{y}) = \mathbf{x}\,\text{Logm}(\mathbf{x}^{-1}\mathbf{y}) \quad , \tag{44}$$



(a) (b) (c)

Fig. 4 Results on synthetic data. (a) The errors of the different filters as a function of the dimensionality of the state space ($\gamma_{\text{state}} = 0.2$ and $\gamma_{\text{obs}} = 0.1$). (b) The runtime of the different filters as a function of the dimensionality of the state space. (c) The errors of the different filters as a function of the state noise.

where Logm$(\cdot)$ is the matrix logarithm which is defined as the inverse of Expm. Both Expm and Logm have efficient numerical implementations, which are readily available with most numerical programming environments [47].

Inspired by Porikli et al. [33], we build a tracker using covariance features as observations. We use pixel positions, RGB colours and the norm of the intensity gradient as our basic features. We let the state $\mathbf{x}_t = (x_t, y_t, r_t)^T \in \mathbb{R}^3$ consist of the position and radius of the object being tracked, and let $\mathbf{C}_1$ denote the covariance feature of the tracked region in the first frame. Essentially, Porikli et al. [33] then compute covariance features in the neighbourhood of the current state and update the state to the position and radius of the best matching $\mathbf{C}_1$ (according to the Riemannian metric). This greedy approach works reasonably well, but lacks a principled way of introducing a smoothing prior. Porikli et al. [33] suggest an *ad hoc* smoothing by introducing a temporal average over a fixed time window.

We introduce smoothing by using the Riemannian UKF for tracking. As the dynamical model we use the identify, $f(\mathbf{x}) = \mathbf{x}$, and for the observation we search locally for the best matching region, just as Porikli et al. [33]. This makes the state space Euclidean and the observation space $\mathcal{M}_{\text{obs}} = \text{Sym}_+$. This tracker uses the same model as Tyagi and Davis [45], but where they use update rules specific to Sym_+, we apply the more generally applicable unscented transform.

4.2.1 Results

We run the tracker on three sequences: one of Richard Feynman giving a lecture, one of people in a raft going down a river and one of a show at a parade. We compare our results to those attained with the approach from Porikli et al. [33]. As we do not have ground truth positions of the tracked objects, we choose motion smoothness as a measure of quality. Specifically, we measure

$$\mathcal{S} = \frac{1}{T} \sum_{t=2}^{T} \|\mathbf{x}_t - \mathbf{x}_{t-1}\| \quad . \tag{45}$$

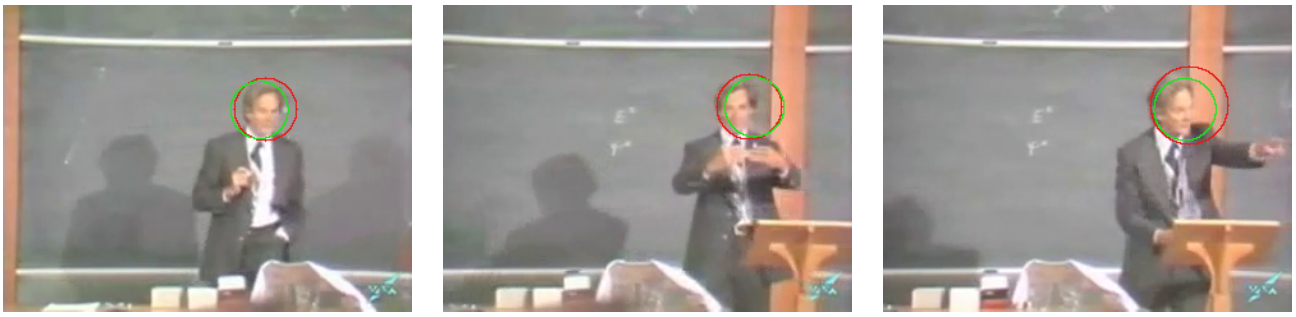
	Riemannian UKF	**Porikli et al. [33]**
Feynman	1.03 ± 0.88	1.53 ± 1.40
Rafting	2.71 ± 1.79	6.73 ± 13.80
Parade	1.73 ± 1.28	3.41 ± 3.23

Table 1 Smoothness of estimated motion paths, eq. 45. Numbers are reported as the mean smoothness $\mathcal{S} \pm$ one standard deviation of the smoothness $\mathcal{S}$.

In fig. 5–7 we show frames from the tracked sequences; movies are available as part of the supplementary material. In general we note that both approaches are able to follow the motion, but the Riemannian UKF produces much more smooth motion estimates. This is not really a surprising result as the main purpose of a Kalman filter is to introduce smoothing. In table 1 we show the measured motion smoothness according to eq. 45 along with the corresponding standard deviations. As can be seen, the Riemannian UKF produces substantially more smooth results than the approach by Porikli et al. [33].

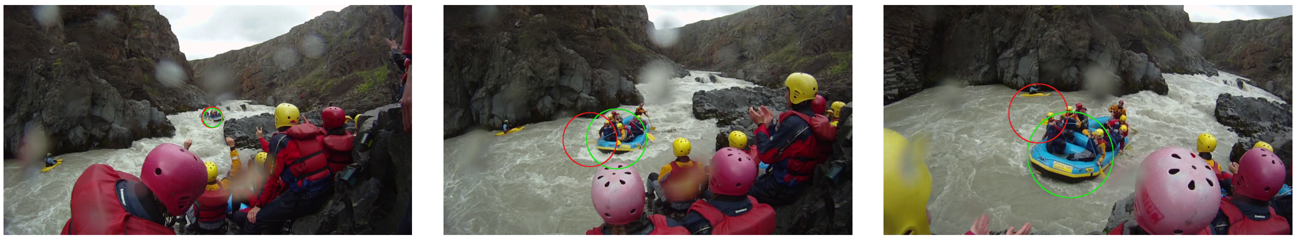
4.3 Articulated Tracking

As a further example, we now build an articulated tracking system using the Riemannian UKF. The objective of such a system is to estimate the pose of a moving person in each frame of an image sequence [32]. As is common [32], we represent human poses using the *kinematic skeleton* (see fig. 8a), which is a collection of rigid bones connected in a tree structure. As the bones have a constant length the joint angles between connected bones are the only degrees of freedom (assuming a fixed root). From joint angles θ, it is trivial to compute bone



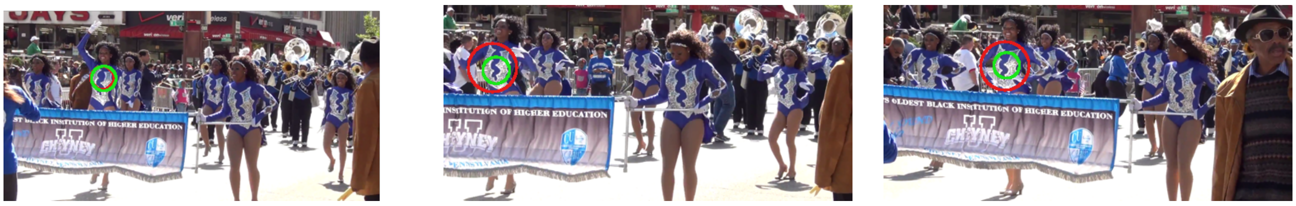
Frame 45 / 530 Frame 342 / 530 Frame 477 / 530

Fig. 5 Selected frames from the `Feynman` sequence; the entire sequence is available in the supplementary material. Results from the approach Porikli et al. [33] is coloured in red, while the Riemannian UKF is in green.



Frame 3 / 216 Frame 160 / 216 Frame 209 / 216

Fig. 6 Selected frames from the `Rafting` sequence; the entire sequence is available in the supplementary material. Results from the approach Porikli et al. [33] is coloured in red, while the Riemannian UKF is in green.



Frame 9 / 354 Frame 163 / 354 Frame 345 / 354

Fig. 7 Selected frames from the `Parade` sequence; the entire sequence is available in the supplementary material. Results from the approach Porikli et al. [33] is coloured in red, while the Riemannian UKF is in green.

end-points using *forward kinematics* [8]. This process starts at the root node of the kinematic tree and recursively rotate and translate bone end-points. This gives a vector $\mathbf{x} = F(\theta) \in \mathbb{R}^{3L}$ containing the three dimensional position of all L bone end-points in the skeleton; here $F(\cdot)$ denotes the forward kinematics function. As bone lengths are constant in this model, $\mathbf{x}$ is confined to parts of $\mathbb{R}^{3L}$, which we denote

$$\mathcal{M} \equiv \{F(\theta) \mid \theta \in \Theta\} \ , \tag{46}$$

where Θ is the space of joint angles. Since the angle space is compact and F is an injective function with a full-rank Jacobian, $\mathcal{M}$ is a compact differentiable manifold embedded in $\mathbb{R}^{3L}$ [15].

The inner product on $\mathcal{M}$ is inherited from $\mathbb{R}^{3L}$ such that the distance measure on $\mathcal{M}$ becomes the length of the physical curves that joints move along [16].

The most common dynamical models for articulated tracking are given by normal distributions in the joint angle space [1, 2, 21, 35]. The metric in this space measures the distance between two poses by looking at the difference between individual joint angles. As a consequence, the movement of an entire arm by a change in the shoulder joint appears as large as the movement of a finger by a change in a finger joint. This rather unnatural metric makes the dynamical model unstable as some limb positions becomes inherently more variant than others. This problem is largely alleviated by designing dynamical models on $\mathcal{M}$ rather than in joint angle space [15, 16].

As $\mathcal{M}$ is a rather complicated manifold, closed-form expressions for the exponential map and parallel transport are not available, so we resort to numerical techniques. We use a forward Euler scheme based on the *standard projection method* for exponential maps and

Schild's Ladder for parallel transports. Detailed descriptions of these standard tools can be found in the literature (e.g. [11, 29]), but a short explanation is also available in appendix B.

4.3.1 Observation Model

To define the filter, we need to be able to generate observations $h(\sigma_{\mathcal{M}}^{(n)})$ corresponding to each sigma point. We use data from a consumer stereo camera[2], which extracts tree dimensional data from images of size 320×240. This gives around 5000 three dimensional points on the body of the person in each frame; we denote these points $\mathbf{Z}_t$. The camera records at roughly 12 frames per second, which makes the dynamical model important.

As the data is a set of three dimensional points, $h(\sigma_{\mathcal{M}}^{(n)})$ should generate a set of comparable three dimensional points. As the camera observes the surface (or skin) of the person we let $h(\sigma_{\mathcal{M}}^{(n)})$ generate a set of three dimensional points on the skin of the pose parameterised by $\sigma_{\mathcal{M}}^{(n)}$. We will generate these points by projecting the actual observed points onto the skin of the pose. This requires a skin model. To simplify the projection, we define the skin of a pose by associating a capsule with each bone (see fig. 8b). The projection of a point onto the skin can then be defined as finding the closest point on the nearest capsule, i.e.

$$\mathrm{proj}_{\mathrm{skin}(\sigma_{\mathcal{M}}^{(n)})}(\mathbf{z}) = \arg\min_{\mathbf{p}_l^{(n)}} \|\mathbf{p}_l^{(n)} - \mathbf{z}\| \tag{47}$$

where $\mathbf{p}_l^{(n)}$ is the point on the capsule associated with the l^{th} bone which is nearest to $\mathbf{z}$ (in the Euclidean sense). This minimisation can trivially be performed by iterating over all bones in the skeleton; see [13] for details.

In summary, we define the generative observation model as

$$h(\sigma_{\mathcal{M}}^{(n)}) = \mathrm{proj}_{\mathrm{skin}(\sigma_{\mathcal{M}}^{(n)})}(\mathbf{Z}_t) \quad . \tag{48}$$

This is the same likelihood system as presented in [13].

4.3.2 Dynamical Model

For the dynamical model, we shall use the simplest of all to predict the motion, i.e.

$$f(\mathbf{x}_{t-1}) = \mathbf{x}_{t-1} \quad , \tag{49}$$

as experience indicates that such models work better than e.g. second order models [1]. This model can easily be expressed in the tangent space such that no logarithm maps are required, which simplifies the numerical implementation substantially.

[2] `http://www.ptgrey.com/products/bumblebee2/`

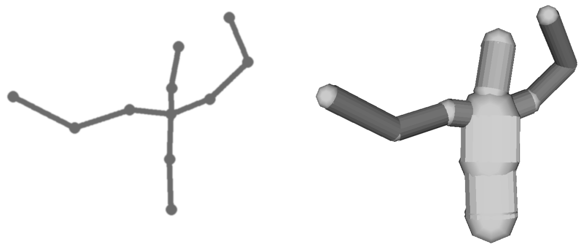


Fig. 8 (a) The kinematic skeleton used for representing poses. (b) The skin model.

4.3.3 Results

We apply the Riemannian UKF and a particle filter with the corresponding motion model — Brownian motion on $\mathcal{M}$ [16] — on two sequences consisting of 300 frames each. Movies are available as part of the supplementary material and a few frames attained using both filters are shown in fig. 11 and 12. In both sequences the person is standing in place and only moving his upper body. We, thus, only model the upper body parts.

In both sequences, the person moves his arms both parallel and orthogonal to the image plane, causing self-occlusions, which pose a challenge for the tracker. The Riemannian UKF is able to successfully track this motion, though some jitter is observed due to the numerical exponential maps and parallel transports. In a few frames the filter looses track, but it quickly recovers. The particle filter, on the other hand, provides visually poorer results.

In the second sequence the person is wearing motion capture markers, which allows us to estimate the tracking error. We use the standard measure corresponding to the average distance between the motion capture markers and the surface of the estimated pose [36],

$$\mathcal{E}(\mathbf{x}_{1:T}) = \frac{1}{TL}\sum_{t=1}^{T}\sum_{l=1}^{L} \|\mathbf{m}_{tl} - \mathrm{proj}_{\mathrm{skin}(\sigma_{\mathcal{M}}^{(n)})}(\mathbf{m}_{tl})\|, \tag{50}$$

where $\mathbf{m}_{tl}$ is the position of the l^{th} marker at time t. The Riemannian UKF achieves an error of 2.7 cm. The error of the particle filter depends on the number of particles, which are plotted in fig. 9. As can be seen, the particle filter requires more than 750 particles to achieve an error comparable to the Riemannian UKF. This, however, requires substantially more likelihood evaluations than the Riemannian UKF, which makes the particle filter use more computation time. In fig. 10 we plot the running time of the two filters[3]. Here it

[3] These results are attained using a single-thread `C++` implementation on a 1.6 GHz Intel Xeon.

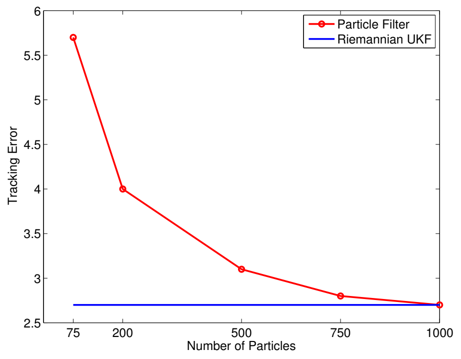


Fig. 9 Tracking error for the Riemannian UKF and the particle filter with varying number of particles.

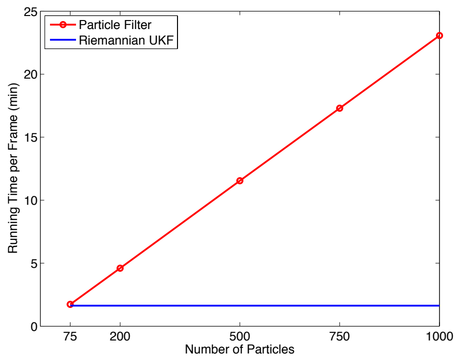


Fig. 10 Running time for the Riemannian UKF and the particle filter with varying number of particles.

shows that the particle filter with 75 particles has approximately the same running time as the Riemannian UKF, whereas the particle filter with 750 particles requires 10 times as long.

4.4 Optimisation on Manifolds

A common problem is that of optimising a function defined over a manifold, e.g. when computing mean values [30] and exact principal components [40]. When the gradient of the cost function is available, standard algorithms, such as *gradient descent* and *simulated annealing* are readily available [27]. In many practical scenarios it can, however, be difficult or computationally prohibitive to acquire such gradients, and stochastic approaches akin to the particle filter have to be used [26].

In the Euclidean setting, it is well-known that the extended Kalman filter can be viewed as a single step in a *Gauss Newton* optimisation scheme [3]. Both the extended and the unscented Kalman filter have been used to solve optimisation problems, such as weight learning in neural networks [37, 46]. This has the advantage that the gradient of the cost function is not needed.

In this section we empirically investigate if the approach carries through to the Riemannian setting on two examples.

4.4.1 Estimating the Karcher Mean

First, we consider the optimisation problem of estimating the Karcher mean (eq. 6) of a set of points on a sphere. This allows us to compare the estimated mean with the ground truth as this can be computed in closed-form,

$$\mu(\mathbf{x}_{1:N}) = \frac{\sum_{n=1}^{N} \mathbf{x}_n}{\| \sum_{n=1}^{N} \mathbf{x}_n \|} \ . \tag{51}$$

To evaluate the potential of using the Riemannian UKF as an optimisation algorithm, we use it for computing the mean value of a set of synthetically generated data points. We generate these by sampling Gaussian data in the tangent space of a sphere and move these to the sphere using the exponential map. This gives us data like those shown in fig. 13. We then compute the minima of eq. 6 using the Riemannian UKF described below. We should stress that better algorithms exist for computing mean values on manifolds (see e.g. [30]) and that we only use this problem as an example of *gradient-free* optimisation on manifolds.

We define the motion model as the identity, i.e. $f(\mathbf{x}) = \mathbf{x}$, using a fixed diagonal covariance[4]. We define the observation model to generate the vector containing the differences between the data and the sigma point measured in the tangent space in the mean estimate μ_{t-1} from the previous iteration, i.e.

$$h(\sigma_{\mathcal{M}}) = \mathrm{Log}_{\mu_{t-1}}(\sigma_{\mathcal{M}}) - \mathrm{Log}_{\mu_{t-1}}(\mathbf{x}_{1:N}) \ . \tag{52}$$

With these definitions, we can use the Riemannian UKF as an optimisation algorithm. We initialise the optimisation by randomly sampling one data point. In fig. 14 we show the distance between the ground truth mean value (eq. 51) and the current estimate of the mean as a function of the number of iterations. We repeat this experiment for several random initialisations, and consistently observe that the algorithm converges near the true mean (distance to true mean is below 10^{-4}). We, however, never observe that the algorithm finds the *exact* mean even after thousands of iterations. We speculate that this is because the algorithm does not incorporate any *line search* strategy.

[4] We also experimented with slowly decreasing the size of this covariance, akin to *weight decay*, but did not observe any noticeable difference for this problem.

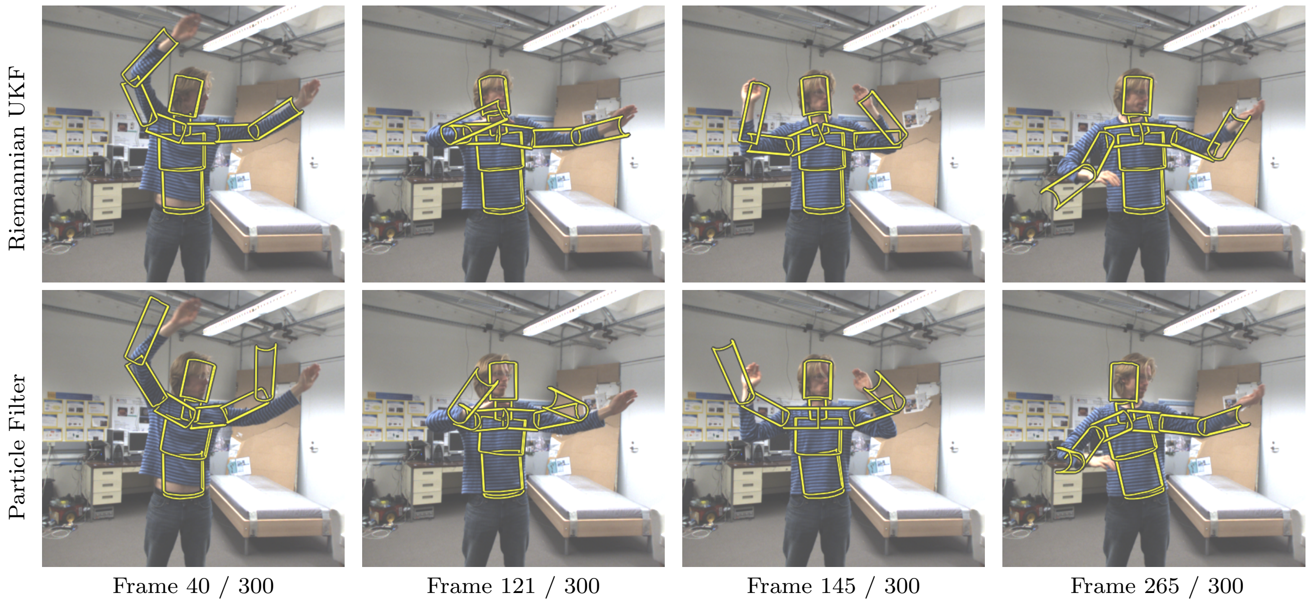


Fig. 11 Four frames from the first sequence; the entire film is available in the supplementary material. The particle filter used 200 particles, which makes it around 3 times as expensive as the Riemannian UKF.

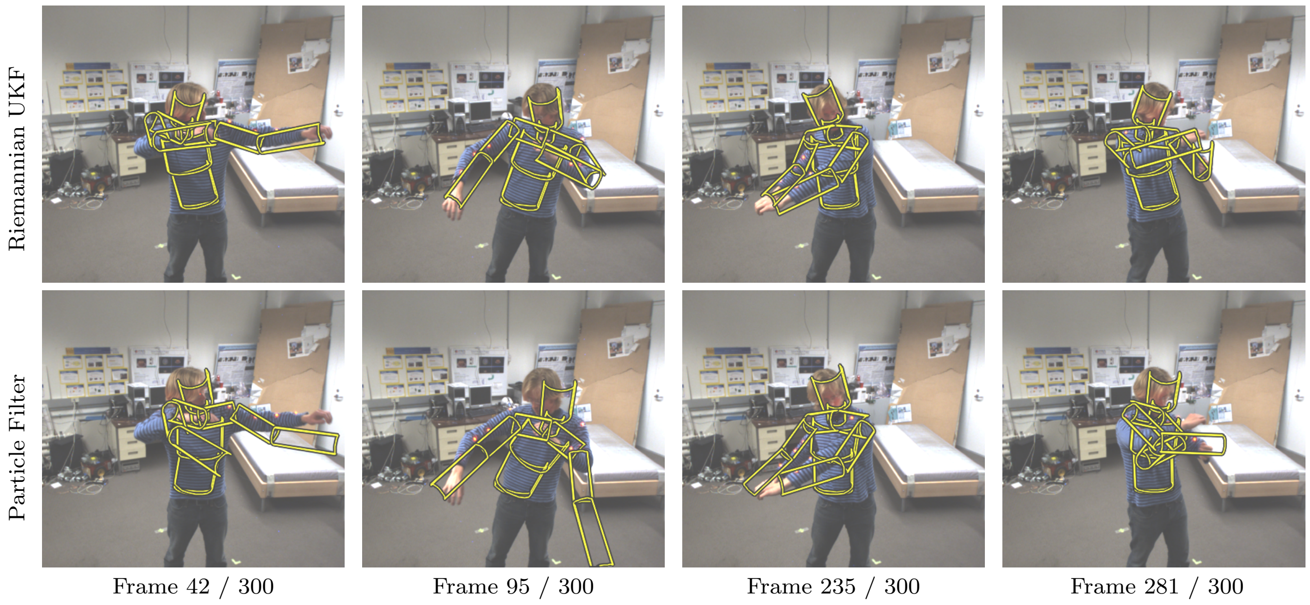


Fig. 12 Four frames from the second sequence; again, the entire film is available in the supplementary material. The particle filter used 200 particles, which makes it around 3 times as expensive as the Riemannian UKF.

4.4.2 Pose Fitting

As a more realistic optimisation problem, we consider a pose fitting problem related to the previously studied articulated tracking problem. We position a skeleton far away from the true pose (see fig. 15a) and optimise the likelihood used in the tracking example by repeated iterations of the unscented Kalman filter. After approximately 60 iterations this converges to the correct pose (see fig. 15b). A video showing the iterations are available in the supplementary material and the optimised error measure is shown in fig. 15c. As in the previous example, we observe that the algorithm "jumps around" the located mode. As in the articulated tracking example, part of this problem is due to numerical inaccuracies in the exponential maps and parallel transports, but we believe that the problem could be mitigated by incorporating a line search strategy into the algorithm.

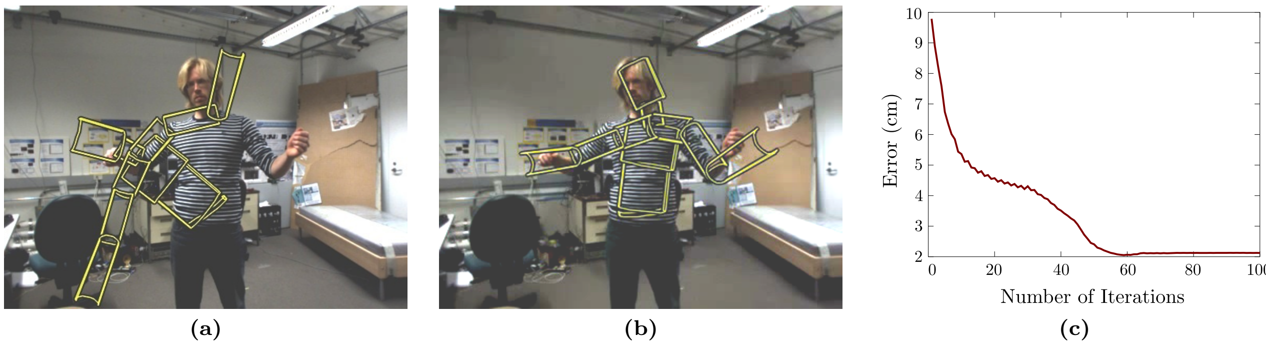


Fig. 15 Using the UKF as an optimisation scheme on manifolds. (a) The initialisation of the optimisation scheme. (b) The located optimum. (c) The error measure of the optimisation as a function of the number of iterations.

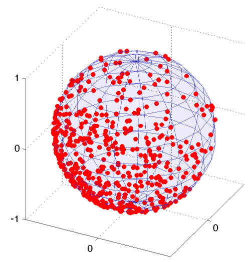


Fig. 13 Synthetic data used for studying convergence. The data is generated by sampling from a Gaussian in the tangent of the sphere; the data is moved to the manifold using the exponential map. The Gaussian in the tangent space has a standard deviation of 0.75.

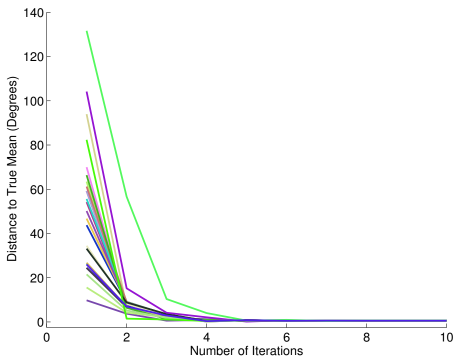


Fig. 14 Convergence of the mean value optimisation. Each plot shows the distance to the true mean as a function of the number of iterations. Each optimisation is initialised by picking a random data point; we have never observed a diverging optimisation.

5 Conclusion

In this paper, we have introduced an extension of the unscented transform and the unscented Kalman filter to Riemannian manifolds. The idea of working with sigma points seems to be a perfect fit for the Riemannian extension as the approach is both practical and gives descriptive results. The suggested filter has the advantage that only limited knowledge of the manifold is needed for an implementation: in the most general case, only the exponential map, the logarithm map and the parallel transport are required. These are available in closed-form for simple manifolds and numerical techniques exists for more complex scenarios. This makes the filter readily applicable for a wide range of problems. We have successfully illustrated the filter on several different problems using different manifold structures, which highlights the generality of the approach.

An interesting direction for future research is to use the Riemannian UKF as a proposal distribution in a particle filter. In Euclidean spaces, this strategy has proven itself highly useful under the name *the unscented particle filter* [28]. It would be interesting to see if this strategy generalises to the Riemannian domain.

Besides tracking applications, we have also empirically shown how the filter can be used as a general-purpose gradient-free optimisation scheme on manifolds. At this stage we do not have proofs of convergence for this strategy, but the empirical results are encouraging. As the filter is fairly easy to implement, it can be applied in many interesting scenarios. This is, however, left for future work.

Acknowledgements: Søren Hauberg would like to thank the Villum Foundation for financial support.

References

1. Balan AO, Sigal L, Black MJ (2005) A Quantitative Evaluation of Video-based 3D Person Tracking. In: 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pp 349–356
2. Bandouch J, Engstler F, Beetz M (2008) Accurate Human Motion Capture Using an Ergonomics-Based Anthropometric Human Model. In: AMDO '08: Proceedings of the 5th international conference on Articulated Motion and Deformable Objects, Springer-Verlag, Lecture Notes in Computer Science (LNCS), vol 5098, pp 248–258

3. Bell BM, Cathey FW (1993) The Iterated Kalman Filter Update as a Gauss-Newton Method. IEEE Transactions on Automatic Control 38:294–297
4. Cappé O, Godsill SJ, Moulines E (2007) An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo. Proceedings of the IEEE 95(5):899–924
5. do Carmo MP (1992) Riemannian Geometry. Birkhäuser Boston
6. Caselles V, Kimmel R, Sapiro G (1997) Geodesic Active Contours. International Journal of Computer Vision 22:61–79
7. Engell-Nørregård M, Erleben K (2011) A projected back-tracking line-search for constrained interactive inverse kinematics. Computers & Graphics 35(2):288–298
8. Erleben K, Sporring J, Henriksen K, Dohlmann H (2005) Physics Based Animation. Charles River Media
9. Fletcher PT, Joshi S (2007) Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data. Signal Processing 87:250–262
10. Fletcher PT, Lu C, Pizer SM, Joshi S (2004) Principal Geodesic Analysis for the Study of Nonlinear Statistics of Shape. IEEE Transactions on Medical Imaging 23(8):995–1005
11. Hairer E, Lubich C, Wanner G (2004) Geometric Numerical Integration: Structure Preserving Algorithms for Ordinary Differential Equations. Springer
12. Hauberg S, Pedersen KS (2010) Stick It! Articulated Tracking using Spatial Rigid Object Priors. In: Asian Conference on Computer Vision, Springer, Lecture Notes in Computer Science, vol 6494
13. Hauberg S, Pedersen KS (2011) Predicting Articulated Human Motion from Spatial Processes. International Journal of Computer Vision 94:317–334
14. Hauberg S, Pedersen KS (2012) HUMIM Software for Articulated Tracking. Tech. Rep. 01/2012, Department of Computer Science, University of Copenhagen
15. Hauberg S, Sommer S, Pedersen KS (2010) Gaussian-like Spatial Priors for Articulated Tracking. In: ECCV, Springer, Lecture Notes in Computer Science, vol 6311, pp 425–437
16. Hauberg S, Sommer S, Pedersen KS (2012) Natural Metrics and Least-Committed Priors for Articulated Tracking. Image and Vision Computing 30(6–7):453–461
17. Julier SJ, Uhlmann JK (1997) A New Extension of the Kalman Filter to Nonlinear Systems. In: International Symposium Aerospace/Defense Sensing, Simulation and Controls, pp 182–193
18. Kalman R (1960) A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME-Journal of Basic Engineering 82(D):35–45
19. Karcher H (1977) Riemannian Center of Mass and Mollifier Smoothing. Communications on Pure and Applied Mathematics 30(5):509–541
20. Kendall DG (1984) Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces. Bulletin of the London Mathematical Society 16(2):81–121
21. Kjellström H, Kragić D, Black MJ (2010) Tracking People Interacting with Objects. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 747–754
22. Kraft E (2003) A Quaternion-based Unscented Kalman Filter for Orientation Tracking. In: Proceedings of the Sixth International Conference on Information Fusion, pp 47–54
23. Kwon J, Lee KM (2010) Monocular SLAM with Locally Planar Landmarks via Geometric Rao-Blackwellized Particle Filtering on Lie Groups. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1522–1529
24. Kwon J, Lee KM, Park FC (2009) Visual Tracking via Geometric Particle Filtering on the Affine Group with Optimal Importance Functions. In: Computer Vision and Pattern Recognition, pp 991–998
25. Lewis FL (1986) Optimal Estimation: With an Introduction to Stochastic Control Theory. Wiley
26. Li R, Chellappa R (2010) Aligning Spatio-Temporal Signals on a Special Manifold. In: ECCV, Springer, LNCS, vol 6315, pp 547–560
27. Liu X, Srivastava A, Gallivan K (2004) Optimal Linear Representations of Images for Object Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(5):662–666
28. van der Merwe R, Doucet A, Freitas ND, Wan E (2001) The Unscented Particle Filter. In: Advances in Neural Information Processing Systems (NIPS 2000), MIT Press, vol 13, pp 584–590
29. Misner C, Thorne K, Wheeler J (1973) Gravitation. W. H. Freeman and Company
30. Pennec X (2006) Intrinsic statistics on riemannian manifolds: Basic tools for geometric measurements. Journal of Mathematical Imaging and Vision 25(1):127–154
31. Pennec X, Fillard P, Ayache N (2004) A Riemannian Framework for Tensor Computing. International Journal of Computer Vision 66:41–66
32. Poppe R (2007) Vision-based human motion analysis: An overview. Computer Vision and Image Understanding 108(1-2):4–18

33. Porikli F, Tuzel O, Meer P (2006) Covariance Tracking using Model Update Based on Lie Algebra. In: Computer Vision and Pattern Recognition, vol 1, pp 728 – 735
34. Rabiner LR (1989) A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: Proceedings of the IEEE, pp 257–286
35. Sidenbladh H, Black MJ, Fleet DJ (2000) Stochastic tracking of 3D human figures using 2D image motion. In: ECCV, Springer, Lecture Notes in Computer Science 1843, vol II, pp 702–718
36. Sigal L, Black MJ (2007) HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion. Tech. Rep. CS-06-08, Brown University
37. Singhal S, Wu L (1989) Training Multilayer Perceptrons with the Extended Kalman Algorithm. In: Advances in Neural Information Processing Systems 1, pp 133–140
38. Sipos BJ (2008) Application of the Manifold-Constrained Unscented Kalman Filter. In: Position, Location and Navigation Symposium, IEEE/ION, pp 30–43
39. Sommer S, Tatu A, Chen C, Jørgensen DR, de Bruijne M, Loog M, Nielsen M, Lauze F (2009) Bicycle chain shape models. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE Computer Society, pp 157–163
40. Sommer S, Lauze F, Nielsen M (2010) The Differential of the Exponential Map, Jacobi Fields and Exact Principal Geodesic Analysis. CoRR abs/1008.1902
41. Srivastava A, Klassen E (2004) Bayesian and Geometric Subspace Tracking. Advances in Applied Probability 36(1):43–56
42. Subbarao R, Meer P (2009) Nonlinear Mean Shift over Riemannian Manifolds. International Journal of Computer Vision 84(1):1–20
43. Tidefelt H, Schön TB (2009) Robust Point-Mass Filters on Manifolds. In: Proceedings of the 15th IFAC Symposium on System Identification (SYSID), pp 540–545
44. Tuzel O, Porikli F, Meer P (2006) Region Covariance: A Fast Descriptor for Detection and Classification. In: Leonardis A, Bischof H, Pinz A (eds) ECCV, Springer Berlin / Heidelberg, Lecture Notes in Computer Science, vol 3952, pp 589–600
45. Tyagi A, Davis JW (2008) A Recursive Filter for Linear Systems on Riemannian Manifolds. In: Computer Vision and Pattern Recognition, pp 1–8
46. Wan EA, van der Merwe R (2002) The Unscented Kalman Filter for Nonlinear Estimation. In: Adaptive Systems for Signal Processing, Communications, and Control Symposium, IEEE, pp 153–158
47. Ward RC (1977) Numerical Computation of the Matrix Exponential with Accuracy Estimate. SIAM Journal on Numerical Analysis 14:600–610
48. Wu Y, Wu B, Liu J, Lu H (2008) Probabilistic Tracking on Riemannian Manifolds. In: International Conference on Pattern Recognition, pp 1–4

A Definitions from Differential geometry

We give definitions of some concepts from differential geometry that we use in the paper (mainly from [5]) for the convenience of the reader.

1. **Differentiable Manifolds:**
   A differentiable manifold of dimension M is a set $\mathcal{M}$ and a family of injective mappings $\mathcal{T} = \{x_i : U_i \subset \mathbb{R}^M \to \mathcal{M}\}$ of open sets U_i of $\mathbb{R}^M$ into $\mathcal{M}$ such that
   – $\bigcup_i x_i(U_i) = \mathcal{M}$, i.e. the open sets cover $\mathcal{M}$.
   – For any pair i, j with $x_i(U_i) \bigcap x_j(U_j) = W \neq \phi$, the mapping $x_j^{-1} \circ x_i$ is differentiable.
   – The family $\mathcal{T}$ is maximal, which means that if (y, V), $y : V \subset \mathbb{R}^M \to \mathcal{M}$ is such that: for each element of $\mathcal{T}$, (x_i, U_i) with $x_i(U_i) \cap y(V) \neq 0$ implies that $y^{-1} \circ x_i$ is a diffeomorphism, then in fact $(y, V) \in \mathcal{T}$.
2. **Directional derivative of a function along a vector field:**
   A vector field X on $\mathcal{M}$ is a map that associates to each $p \in \mathcal{M}$ an element $X(p) \in T_p\mathcal{M}$, where $T_p\mathcal{M}$ is the tangent space of $\mathcal{M}$ at p. The space of smooth vector fields on $\mathcal{M}$ is denoted $\mathfrak{X}(\mathcal{M})$. Let $f : \mathcal{M} \to \mathbb{R}$ be a differentiable function of $\mathcal{M}$ and X a vector field on $\mathcal{M}$. The directional derivative $X.f$ is the function $\mathcal{M} \to \mathbb{R}$,
   $$(X.f)(p) = df_p(X(p)) \tag{53}$$
   the differential of f at p evaluated at vector $X(p)$.
3. **Covariant tensors:**
   A p-covariant tensor h is a $\mathcal{C}^\infty$ p-linear map
   $$\underbrace{T\mathcal{M} \times \cdots \times T\mathcal{M}}_{p \text{ times}} \to \mathcal{C}^\infty(\mathcal{M}) \tag{54}$$
   i.e., for all $x \in \mathcal{M}$, $x \mapsto h_x$:
   $$v_1, \ldots, v_p \in T_x\mathcal{M} \mapsto h_x(v_1, \ldots, v_p) \in \mathbb{R} \tag{55}$$
   is p-linear and for vector fields $X_1, \ldots, X_p \in \mathfrak{X}$, the map $x \mapsto h_x(X_1(x), \ldots, X_p(x))$ is smooth.
4. **Riemannian Metric:**
   A Riemannian metric on a manifold $\mathcal{M}$ is a covariant 2-tensor g which associates to each point $p \in \mathcal{M}$ an inner product $g_p = \langle -, - \rangle_p$ on the tangent space $T_p\mathcal{M}$, i.e., not only is it bilinear, but symmetric and positive definite and thus define a Euclidean distance on each tangent space. In terms of local coordinates, the metric at each point x is given by a matrix, $g_{ij} = \langle X_i, X_j \rangle_x$, where X_i, X_j are tangent vectors to $\mathcal{M}$ at x, and it varies smoothly with x. A *Geodesic curve* is a local minimiser of arc-length computed with a Riemannian metric.

5. **Affine connection:**
An affine connection ∇ on a differentiable manifold $\mathcal{M}$ is a mapping
$$\nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M}) \tag{56}$$
which is denoted by $\nabla(X,Y) \to \nabla_X Y$ and which satisfies the following properties:
   - $\nabla_{fX+gY} Z = f\nabla_X Z + g\nabla_Y Z$.
   - $\nabla_X(Y+Z) = \nabla_X Y + \nabla_X Z$.
   - $\nabla_X(fY) = f\nabla_X Y + X(f)Y$.

in which $X, Y, Z \in \mathfrak{X}(\mathcal{M})$ and f, g are $\mathcal{C}^\infty(\mathcal{M})$. This gives a notion of directional derivative of a vector field defined on the manifold. An affine connection extends naturally to more than vector fields, and especially of interest here, covariant tensors: if h is a covariant p-tensor and $X \in \mathfrak{X}(\mathcal{M})$, $\nabla_X h$ is defined as follows. Given p vector fields $Y_1, \ldots, Y_p \in \mathfrak{X}(\mathcal{M})$,
$$\begin{aligned}(\nabla_X h)(Y_1,\ldots,Y_p) &= X.\left(h(Y_1,\ldots,Y_p)\right)\\ &\quad - \sum_{i=1}^p h(Y_1,\ldots,\nabla_X Y_i,\ldots,X_p)\end{aligned} \tag{57}$$

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

The covariant derivative extends to covariant tensors via the extension of the connection to them: Given a covariant p-tensor h defined along c and vector fields $U_1, \ldots, U_p$ along c,
$$\begin{aligned}\frac{Dh}{dt}(U_1(t),\ldots,U_p(t)) &= \frac{d}{dt}\left(h_t(U_1(t),\ldots,U_p(t)\right)\\ &\quad - \sum_{i=1}^p h_t\left(U_1(t),\ldots \frac{DU_p}{dt},\ldots,U_p(t)\right)\end{aligned} \tag{58}$$

7. **Parallel transport:**
Given a vector $P \in T_{c(0)}\mathcal{M}$, the differential equation
$$\begin{cases}\frac{DP(t)}{dt} &= 0\\ P(0) &= P\end{cases} \tag{59}$$
admits a unique solution, called the parallel transport of P along c. The induced map $P \mapsto P(t)$ from $T_{c(0)}\mathcal{M}$ to $T_{c(t)}\mathcal{M}$ is a linear isomorphism.
The parallel transport extends to covariant tensors in the same way: Given a p-linear mapping $h : T_{c(0)}\mathcal{M} \times \cdots \times T_{c(0)}\mathcal{M} \to \mathbb{R}$, the differential equation
$$\begin{cases}\frac{Dh(t)}{dt} &= 0\\ h(0) &= h\end{cases} \tag{60}$$
admits a unique solution, called the parallel transport of h along c. As for vectors, the mapping $h \mapsto h(t)$ is a linear isomorphism between $p-linear$ maps on $T_{c(0)}\mathcal{M}$ and $p-linear$ maps on $T_{c(t)}\mathcal{M}$.

8. **Levi-Civita connection:**
Given a Riemannian metric g on the manifold $\mathcal{M}$, there exists a unique affine connection ∇ such that
   - compatibility with the metric:
$$X.g(Y,Z) = g(\nabla_X Y, Z) + g(X, \nabla_X Z) \tag{61}$$
   - symmetry:
$$\nabla_X Y - \nabla_Y X = [X,Y] \tag{62}$$
([X, Y] is the Lie bracket of X and Y).

∇ is the *Levi-Civita connection associated to g*. Note that from the previous items, one has $\nabla_X g = 0$ for any $X \in \mathfrak{X}(\mathcal{M})$ and that the parallel transport in that case is a *linear isometry*.
The compatibility of ∇ and the metric g can be expressed in term of covariant derivatives: if $X(t) = X(c(t)$ and $Y(t) = Y(c(t))$ are two vector fields along the curve c, and D/dt is the covariant derivative along c,
$$\frac{d}{dt}g(X(t),Y(t)) = g\left(\frac{DX(t)}{dt}, Y(t)\right) + g\left(X(t), \frac{DY(t)}{dt}\right). \tag{63}$$

9. **Christoffel symbols:**
In a parametrised manifold, where the curve $c(t)$ is represented as $(x^1(t), \ldots, x^M(t))$, the covariant derivative of a vector field v becomes
$$\frac{Dv}{dt} = \sum_m \left\{ \frac{dv^m}{dt} + \sum_{i,j} \Gamma^m_{ij} v^j \frac{dx^i}{dt} \right\} \frac{\partial}{\partial x_m} \tag{64}$$
where the Γ^m_{ij} are the *coefficients of the connection* also known as the *Christoffel symbols Γ*. In particular, the parallel transport equation above becomes the first-order *linear* system
$$\frac{dv^m}{dt} + \sum_{i,j} \Gamma^m_{ij} v^j \frac{dx^i}{dt} = 0, \quad m = 1 \ldots M. \tag{65}$$
For the Levi-Civita connection associated with the metric g, the corresponding Christoffel symbols are given by
$$\Gamma^m_{ij} = \frac{1}{2}\sum_l \left\{ \frac{\partial}{\partial x_i} g_{jm} + \frac{\partial}{\partial x_j} g_{mi} - \frac{\partial}{\partial x_m} g_{ij} \right\} g^{ml} \tag{66}$$
g_{ij} is the ij^{th} element of the metric, and g^{ij} is the ij^{th} element of its inverse. A curve is geodesic if the covariant derivative of its tangent vector field is zero everywhere on it, which means that a geodesic curve has zero tangential acceleration. Such a curve c satisfies the second order system of ODEs, which, with the above parametrisation becomes
$$\frac{d^2x^m}{dt^2} + \sum_{ij} \Gamma^m_{ij} \frac{dx^i}{dt}\frac{dx^j}{dt} = 0, \quad m = 1 \ldots M. \tag{67}$$

10. **Exponential map:**
The exponential map is a map $\mathrm{Exp} : T\mathcal{M} \to \mathcal{M}$, that maps $v \in T_q\mathcal{M}$ for $q \in \mathcal{M}$, to a point $\mathrm{Exp}_q v$ in $\mathcal{M}$ obtained by going out the length equal to $|v|$, starting from q, along a geodesic which passes through q with velocity equal to $\frac{v}{|v|}$. Given $q \in \mathcal{M}$ and $v \in T_q\mathcal{M}$, and a parametrisation $(x_1, \ldots, x_n)$ around q, $\mathrm{Exp}_q(v)$ can be defined as the solution at time 1 of the above system of ODEs (67)

with initial conditions $(x^m(0)) = q$ and $(\frac{dx^m}{dt}(0)) = v$, $m = 1, \dots, M$. The geodesic starting at q with initial velocity t can thus be parametrised as

$$t \mapsto \operatorname{Exp}_q(tv). \tag{68}$$

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

B Numerical Implementation of Exponential Maps and Parallel Transports

In this appendix, we briefly review some techniques for numerical implementation of exponential Maps and parallel transports. We have applied them in the articulated tracking example in sec. 4.3. As logarithm maps are not used, they will not be described here. It is worth noticing that the numerical techniques presented in the following are easily adapted to other manifolds, though care should be taken in, e.g., step size selection when the manifold is highly curved.

B.1 Computing Exponential Maps

Assume $\mathcal{M}$ is an M-dimensional sub-manifold of $\mathbb{R}^N$ and that the metric is inherited from the standard inner product in $\mathbb{R}^N$. We can then discretise the geodesics in a straight-forward manner. We remind the reader that, given a vector $\mathbf{v}$ in the tangent space of the point $\mathbf{x}_0$, the exponential map seeks a point $\operatorname{Exp}_{\mathbf{x}_0}(\mathbf{v})$ on the geodesic curve starting at $\mathbf{x}_0$ with the same length and direction as $\mathbf{v}$.

We then discretise the geodesics in a straightforward way by applying a standard forward Euler scheme based on the *standard projection method* [11]. The method repeatedly takes a discrete step in the tangent space in the direction encoded by $\mathbf{v}$ followed by a projection back onto the manifold. The latter is necessary as the discrete step will “fall of the manifold” unless $\mathcal{M}$ is flat. This scheme is illustrated in fig. 16. The missing piece is a scheme for projecting a point in the embedding space back onto $\mathcal{M}$. Defining projection of $\hat{\mathbf{x}} \in \mathbb{R}^N$ as finding the nearest point on $\mathcal{M}$ reduces projection to a non-linear least-squares problem [15],

$$\operatorname{proj}_{\mathcal{M}}(\hat{\mathbf{x}}) = \arg\min_{\mathbf{x} \in \mathcal{M}} \left(\|\mathbf{x} - \hat{\mathbf{x}}\|^2 \right) \quad , \tag{69}$$

which can easily be solved using gradient descent [15]. Specifically, we apply a projected steepest descent with line-search, as empirical results have shown it to be both fast and stable [7]. This scheme usually finds a local optimum within a few iterations as it is warm-started with the results from the previous iteration.

In the practical implementation, we use 10 discrete steps in the Euler scheme, where the step length is controlled by the length of the tangent vector and the length of the geodesic segment computed so far. More sophisticated schemes that take local curvature into account might prove beneficial, but we have not experimented with this. We have tried with more discrete steps, but did not notice much improvement; we speculate that this is because we only need to compute short geodesic segments due to the sequential nature of the tracking problem.

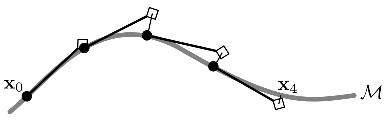


Fig. 16 An illustration of a 4-step *standard projection method* for approximating the exponential map.

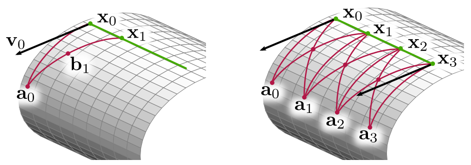


Fig. 17 An illustration of *Schild’s Ladder* for approximating the parallel transport.

B.2 Computing the Parallel Transport

Given two points $\mathbf{x}_0$ and $\mathbf{x}_I$ in $\mathcal{M}$ and the geodesic segment α that joins them, we describe a classical approximation of the parallel transport of a vector $\mathbf{v}_0 \in T_{\mathbf{x}_0}\mathcal{M}$ to a vector $\mathbf{v}_i \in T_{\mathbf{x}_I}\mathcal{M}$ along α, known as *Schild’s Ladder* [29]. This scheme places points along the geodesic and approximately parallel transports $\mathbf{v}_0$ to these by forming approximate parallelograms on $\mathcal{M}$.

Let $\{\mathbf{x}_1, \dots, \mathbf{x}_{I-1}\}$ denote points along the geodesic segment joining $\mathbf{x}_0$ and $\mathbf{x}_I$. Then start by computing $\mathbf{a}_0 = \operatorname{Exp}_{\mathbf{x}_0}(\mathbf{v}_0)$ and the midpoint $\mathbf{b}_1$ of the geodesic segment joining $\mathbf{x}_1$ and $\mathbf{a}_0$ (see the left side of fig. 17). Follow the geodesic from $\mathbf{x}_0$ through $\mathbf{b}_1$ for twice its length to the point $\mathbf{a}_1$. This scheme is repeated for all sampled points along the geodesic from $\mathbf{x}_0$ to $\mathbf{x}_I$ (see the right side of fig. 17). The final parallel transport of $\mathbf{v}_0$ can then be evaluated as the logarithm map $\operatorname{Log}_{\mathbf{x}_I}(\mathbf{a}_I)$. When sampled points are available along the geodesic segment from $\mathbf{x}_I$ to $\mathbf{a}_I$, this logarithm map can easily be approximated using the finite difference of the velocity at $\mathbf{x}_I$.

As we use the standard projection method for computing the geodesic which we transport along, we form approximate parallelograms between the discrete points on the geodesic. As with the exponential maps we do not seem to experience numerical problems caused by curvature of the manifold, most likely due to the short geodesics that occur as part of tracking. For more difficult parallel transportation problems, we believe that more sophisticated numerical schemes are needed.

C Proof of Proposition 1

We need to show that $\frac{DM_t}{dt} = 0$. It will be sufficient to show that

$$\left(\frac{DM_t}{dt} \right) (v_m(t), v_{m'}(t)) = 0, \quad m, m' = 1, \dots, M$$

for the vectors $v_m(t), m = 1, \dots, M$ defined in the proposition. Indeed, for each t, they form an orthonormal basis of $T_{\alpha(t)}\mathcal{M}$:

$$\frac{d}{dt} g\left(v_m(t), v'_m(t)\right) = g\left(\frac{Dv_m}{dt}, v'_m(t)\right) + g\left(v_m(t), \frac{Dv_{m'}}{dt}\right) = 0 \quad (70)$$

because the $v_m(t)$s are parallel along α, and therefore their covariant derivative vanish. By definition of the covariant derivative for a tensor,

$$\left(\frac{DM_t}{dt}\right)\left(v_m(t), v_{m'}(t)\right) = \frac{d}{dt}\left(M_t\left(v_m(t), v_{m'}(t)\right)\right. \quad (71)$$

$$-M_t\left(\frac{Dv_m}{dt}, v_{m'}(t)\right) - M_t\left(v_m(t), \frac{Dv_{m'}}{dt}\right). \quad (72)$$

The last two terms vanish, again because the $v_m(t)$s are parallel. On the other hand, a simple calculation gives

$$M_t\left(v_m(t), v_{m'}(t)\right) = \lambda_m \delta_{mm'} \quad (73)$$

($\delta_{mm'}$ is the usual Kronecker symbol) and this quantity is thus independent of t. This concludes the proof.