AN EFFICIENT PARTICLE FILTERING TECHNIQUE ON THE GRASSMANN MANIFOLD

Quentin Rentmeesters, P.-A. Absil, Paul Van Dooren *

Université catholique de Louvain
B-1348 Louvain-la-Neuve, Belgium

Kyle Gallivan, Anuj Srivastava

Florida State University
Tallahassee, FL, USA 32306

ABSTRACT

Subspace tracking methods are widespread in signal and image processing. To reduce the influence of perturbations or outliers on the measurements, some authors have used a stochastic piecewise constant velocity model on the Grassmann manifold. This paper presents an efficient way to simulate such a model using a particular representation of the Grassmann manifold. By doing so, we can reduce the spatial and time complexity of filtering techniques based on this model. We also propose an approximation of this system which can be computed in a finite number of operations and show similar results if the subspace variation is slow.

Index Terms— time-varying subspace learning, Grassmann manifold, particle filtering

1. INTRODUCTION

Subspace tracking problems appear in many areas such as signal processing and image processing. For instance, in direction of arrival (DOA) tracking, a subspace tracker is required to recursively update the dominant subspace of the estimated covariance matrix of the signal. In image processing, subspace tracking problems appear in moving object tracking on video. In this problem the tracked object is often represented by a linear subspace. This subspace must be updated to cope with change in illumination or viewing angles. Furthermore, to reduce the influence of perturbations or outliers in the measurements, a filter is required. To implement this filter, some authors [1, 2] work on the Grassmann manifold denoted $\mathrm{G}(n, p)$ which is the set of p-dimensional subspaces of R^n and they have introduced a stochastic piecewise geodesic (constant velocity) model as a motion model for the subspaces to reduce the influence of the noise.

In [1], the authors use this model and a particle filtering technique on a DOA tracking problem. This technique approximates the posterior distribution of the subspace at each time step by drawing a large number of samples according to their stochastic piecewise geodesic model. Then, they estimate the subspace by computing the mean of these samples. In [2], the authors use the same model and a Kalman Filter approach on the tangent space of the Grassmann manifold to track an object on a video. Both papers show that better results are obtained using this geometric approach. But these techniques suffer from the high computational cost of the exponential mappings, parallel transports and log-mappings. In fact, the method presented in [1] requires computing the exponential of an $n \times n$ matrix, which is not efficient if n is large.

In this paper, we propose a different implementation of the particle filter introduced in [1], using a computationally more efficient parameterization of the tangent space to the Grassmann manifold yielding a computational complexity of $\mathcal{O}(np^2)$. To further reduce the complexity, we also propose to replace the exponential mapping by a retraction and the parallel transport by a vector transport as introduced in [3]. We show that this can reduce the computational cost while preserving the accuracy if the subspaces vary slowly.

The paper is organized as follows. Section 2 introduces the notation and explains the main principle of our approach. Section 3 describes our implementation of the particle filtering technique. The approximation of the geodesic process using retractions and vector transports is presented in section 4. Numerical simulations are presented in section 5 and section 6 concludes the paper.

2. PRINCIPLE OF OUR APPROACH

In this section, we describe the main idea behind our complexity reduction. We first introduce some notation and recall the definition of a stochastic geodesic process.

A point $\mathcal{X}$ on the Grassmann manifold is represented as the column space of an orthogonal matrix $X \in R^{n \times p}$; we write $\mathcal{X} = \mathrm{col}(X)$. We restrict ourselves to orthogonal matrices to obtain simpler formulas and to avoid conditioning problems.

In the particle filtering algorithm of [1], the authors use the following state space model:

*This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors.

$$[X_k|X_{\perp,k}] = [X_{k-1}|X_{\perp,k-1}] \exp\left(\begin{bmatrix} 0 & -A_{k-1}^\top \\ A_{k-1} & 0 \end{bmatrix}\right), \tag{1}$$

$$A_k = A_{k-1} + N_k, \tag{2}$$

and the observation model:

$$p(Y_k|X_k) = K_k \exp \frac{tr(X_k X_k^\top Y_k Y_k^\top)}{\sigma_{\text{data}}^2}, \tag{3}$$

where $X_k \in R^{n\times p}$, $X_{\perp,k} \in R^{n\times(n-p)}$ such that $[X_k|X_{\perp,k}]$ is orthogonal, $A \in R^{(n-p)\times p}$ and N_k is an $(n-p)\times p$ matrix whose elements are i.i.d. real normals of mean 0 and variance σ_{model}^2. $\mathcal{Y}_k = \text{col}(Y_k)$ are the measurements, σ_{data}^2 is their variance and K_k is a normalizer. Notice that the observation distribution (3) is independent of the particular orthogonal basis we have chosen to represent X_k and Y_k.

The state space model (1-2) defines a piecewise-geodesic model where the velocities on all pieces are related by a Markov process. This state space model is used in [1] to simulate the movement of particles; in [2] the same model is used to implement a Kalman filter but the velocity model represented by the matrix A is different. Therefore, it is very important to perform simulations with this model at a very low cost. Implementing the model as in (1-2), may yield a dramatically suboptimal spatial complexity. To see this, observe that the dimension of the Grassmann manifold $\mathrm{G}(n,p)$ is $p(n-p)$. So is the dimension of its tangent space at a given point. Hence, a position-velocity pair on $\mathrm{G}(n,p)$ involves $2p(n-p)$ degrees of freedom. Yet, the data available at time k for one particle in (1) is $(X_k, X_{\perp,k}, A_k)$ which involves $np + n(n-p) + (n-p)p = n^2 + (n-p)p$ real numbers. When $n >> p$ this is much larger than the theoretical optimum of $2p(n-p)$ parameters.

It would be possible to achieve the theoretical optimal number of parameters by working in local parameterizations of the Grassmann manifold. This option has drawbacks: the equations are less streamlined, and the transition between local parameterizations has to be managed.

In [4], a method is proposed to compute the exponential mapping in $\mathcal{O}(np^2)$ flops. But, at time k, only X_k is computed, and an orthogonal completion of X_k is built based on Householder vectors. This orthogonal completion is not equivalent to $X_{\perp,k}$ defined in (1). So at the next time step we cannot compute X_{k+1} because $X_{\perp,k}$ is required. Therefore, this approach does not solve the problem in this case.

In this paper, we propose to represent the state $(\mathcal{X}_k, \dot{\mathcal{X}}_k)$ of a particle at time k by (X_k, V_k), where $X_k \in R^{n\times p}$ is such that $\text{col}(X_k) = \mathcal{X}_k$ and $V_k \in R^{n\times p}$ is such that $X_k^\top V_k = 0$ and $\frac{d}{dt}\text{col}(X_k + tV_k)|_{t=0} = \dot{\mathcal{X}}_k$ is the velocity of the particle. It can be shown that, given X_k, this V_k is unique and moreover, if X_k is replaced by $X_k Q$, then V_k is replaced by $V_k Q$ for any orthogonal $Q \in R^{p\times p}$. Notice also that V_k is related to A_k in (2) by $V_k = X_{k,\perp} A_k$, see [5].

Observe that our representation uses $2np$ real variables, which is smaller than $n^2 + (n-p)p$ for $1 \le p < \frac{n}{2}$ and $n > 1$ and is particularly interesting when $n >> p$.

3. REDUCED-MEMORY IMPLEMENTATION OF PARTICLE FILTERING

In this section, we explain how the subspace tracking method introduced in [1] can be implemented in $\mathcal{O}(np^2)$ flops using the representation introduced in the previous section.

The following formulas [5], are equivalent to the state space model (1-2) to update X_k and V_k:

$$V_{k-1} = U_{k-1}\Sigma_{k-1}W_{k-1}^\top \text{ (compact svd)}, \tag{4}$$

$$D_{k-1} = X_{k-1}W_{k-1}, \tag{5}$$

$$X_k = \left(D_{k-1}\cos\Sigma_{k-1} + U_{k-1}\sin\Sigma_{k-1}\right) {W_{k-1}}^\top, \tag{6}$$

$$\begin{aligned} V_k = &\left(-D_{k-1}\sin\Sigma_{k-1} + U_{k-1}\cos\Sigma_{k-1}\right)\Sigma_{k-1}{W_{k-1}}^\top \\ &+ (\Omega_k - X_k {X_k}^\top \Omega_k). \end{aligned} \tag{7}$$

Equation (6) can be interpreted as the computation of the geodesic and equation (7) as a parallel transport of the velocity vector V_{k-1} at X_{k-1} along the geodesic curve between $\mathcal{X}_{k-1}$ and $\mathcal{X}_k$ where $\mathcal{X}_k$ is represented by X_k. From a statistical point of view, equation (7) is equivalent to (2). Using the fact that $V_k = X_{\perp,k}A_k$, the perturbation in (7) becomes: $(I_n - X_k X_k^\top)\Omega_k = X_{\perp,k}X_{\perp,k}^\top\Omega_k = X_{\perp,k}\tilde{\Omega}_k$ where $\tilde{\Omega}_k = X_{\perp,k}^\top \Omega_k$ is also a Gaussian noise with the same variance than Ω_k because $X_{\perp,k}^\top$ is orthogonal. Thus, this $\tilde{\Omega}_k$ corresponds to the N_k in (2). Notice also that this way of computing the geodesic model requires $\mathcal{O}(np^2)$ flops per step, the most computationally demanding step being the SVD.

In [1], an extrinsic version of the mean on $\mathrm{G}(n,p)$ is used. This mean is the dominant p-dimensional eigenspace of $\tilde{G}_k = \frac{1}{M}\sum_{i=1}^M X_k^i {X_k^i}^\top$ and it requires the computation of an SVD of the $n \times n$ matrix $\tilde{G}_k$ which costs $\mathcal{O}(n^3)$ flops. To reach our computational complexity of $\mathcal{O}(np^2)$, this extrinsic mean can be replaced by the Karcher mean:

$$\mu(X^1,...,X^M) = \arg\min_{Y\in\mathrm{G}(n,p)} \frac{1}{2M}\sum_{i=1}^M D(X^i,Y)^2,$$

where $D(X^i, Y)$ stands for the geodesic distance between X^i and Y. There is no closed form formula to compute this mean. One way to solve this minimization problem is to implement a gradient descent method as follows:

1. Compute $\tilde{V} = \frac{1}{M}\sum_{i=1}^M \log_{\mu^k}(X^i)$ where $\log_X(Y)$ stands for the log-mapping, i.e., it returns the velocity vector such that we get Y if we follow for unit time the constant speed geodesic starting at X with speed

n	10	20	50	100
# iterations	6	7	9	13

Table 1. Number of iterations versus problem size for 1000 points on $\mathrm{G}(n, \frac{n}{5})$ to reach $\|\tilde{V}\|_F < 10^{-6}$.

given by the norm of this velocity vector in the direction of this velocity vector. The method proposed in [4] to compute this log-mapping using a CS decomposition can be slightly modified to work with the parameterization of the Grassmann manifold introduced in section 2:

$$\begin{bmatrix} X^\top Y \\ (I_n - XX^\top)Y \end{bmatrix} = \begin{bmatrix} W_1 \cos(\Sigma) Z^\top \\ W_2 \sin(\Sigma) Z^\top \end{bmatrix},$$
$$\log_X(Y) = W_2 \Sigma W_1^\top.$$

2. Move along a geodesic curve in the direction $\tilde{V} = U\Sigma W^\top$: set $\mu^{k+1} = \left(\mu^k W \sin(\Sigma) + U\cos(\Sigma)\right) W^\top$ and iterate.

This method requires $\mathrm{O}(Mnp^2)$ flops per iteration and the number of iterations required to reach a given accuracy appears to grow less than linearly with n and p as Table 1 shows. Moreover, only a few iterations need to be computed if an high precision is not required. It is thus more efficient than the extrinsic mean when n is large.

4. FURTHER COMPLEXITY REDUCTION BY APPROXIMATION

One way to further reduce the computational cost is to replace the geodesics by retractions and the parallel transports by vector transports. This makes it possible to replace the SVD decomposition of an $n \times p$ matrix by a QR decomposition which can be performed in a finite number of steps. This is particularly interesting if p is close to $\frac{n}{2}$.

More precisely, we propose to replace the piecewise constant velocity model (4-7) by the following process:

$$X_k = q(X_{k-1} + V_{k-1}), \tag{8}$$

$$V_k = \frac{(I_n - X_k {X_k}^\top) V_{k-1} \|V_{k-1}\|_F}{\|(I_n - X_k {X_k}^\top) V_{k-1}\|_F} + (I_n - X_k {X_k}^\top)\Omega_k, \tag{9}$$

where q is a projection of an $n \times p$ matrix of full rank onto the Stiefel manifold $\mathrm{St}(n,p)$ (the set of orthogonal $n \times p$ matrices) such that $\mathrm{col}(X) = \mathrm{col}(q(X))$ and $q(W) = W \; \forall W \in \mathrm{St}(n,p)$ and $\|.\|_F$ denotes the Frobenius norm. This system can be thought as a first order approximation of the geodesic process projected on the Stiefel manifold. Therefore, if $\|V_k\|_F$ is small, we obtain a good approximation of the geodesic.

For all $V = U\Sigma W^\top$ such that $\tan(\Sigma)$ exists, it is possible to find $\tilde{V}$ such that (4-6) with $V_{k-1} = V$ and (8) with $V_{k-1} = \tilde{V}$ yield two X_k's that have the same column space, i.e., satisfies

$$\mathrm{col}(\underbrace{(XW\cos(\Sigma) + U\sin(\Sigma))W^\top}_{Y}) = \mathrm{col}(X + \tilde{V}).$$

In fact, we can right multiply Y by the invertible matrix $W\cos(\Sigma)^{-1}W^\top$ (for $\sigma_i < \pi/2$) without changing its column space:

$$\mathrm{col}(Y) = \mathrm{col}(X + \underbrace{U\tan(\Sigma)W^\top}_{\tilde{V}}).$$

Observe that this $\tilde{V}$ is given by:

$$\tilde{V} = Y(X^\top Y)^{-1} - X.$$

Consequently, we can compute $\tilde{V}_0$ using this formula and then use (8-9) with this $\tilde{V}_0$ to simulate the stochastic geodesic process. Observe also that the vector transport (9) is chosen such that $\|V_k\| = \|V_{k-1}\|$ when $\Omega_k = 0$. Notice however that the geodesic process (4-7) is not equivalent to (8-9) from a statistical point of view but close to it if $\|V\|_F$ is sufficiently small ($\tan(\Sigma) \approx \Sigma$ for small Σ).

It remains to choose the projection q in (8). One possibility is to use the polar decomposition which is the nearest element of the Stiefel manifold in terms of the Frobenius norm. This is the best choice. In fact, as we have $(X + \tilde{V}) = Y(W\cos(\Sigma)^{-1}W^\top)$ the orthogonal factor of the polar decomposition of $(X + \tilde{V})$ will be Y since $(W\cos(\Sigma)^{-1}W^\top)$ is a positive definite matrix ($0 < \sigma_i < \pi/2$). This decomposition also satisfies the following property: $\forall\, W \in \mathrm{St}(p,p)$, $q(XW) = q(X)W$ which makes (8-9) invariant to the specific choice of orthogonal basis used to represent the subspaces. More precisely, if X_1 is the first basis and V_1 is the corresponding tangent vector, this means that a change of basis: $X_1 \leftarrow X_1 W$ and $V_1 \leftarrow V_1 W$ for any $W \in \mathrm{St}(p,p)$ will not change the trajectory (on the Grassmann manifold) of (8-9).

Unfortunately, this decomposition is as expensive as an SVD. We therefore propose to use a QR-decomposition (with positive elements on the diagonal of R to satisfy $q(W) = W \; \forall W \in \mathrm{St}(n,p)$). If the subspaces vary slowly, i.e., if $\|V\|_F$ is small, then $X + V$ will be close to an orthogonal matrix and the Q factor of the QR-decomposition will be close to the Q factor of the polar decomposition.

5. EXPERIMENTS AND RESULTS

To compare the gain in terms of computational complexity, the computational time required to simulate a stochastic piecewise geodesic process was computed, see Table 2. The complexity reduction is important on problems where p is

	(A)	(B)	(C)
$n = 100, p = 5$	24 %	7 %	10 %
$n = 100, p = 25$	43 %	34 %	19 %
$n = 100, p = 50$	75 %	97 %	37 %

Table 2. Computational time in percent of the computational time spent by the implementation of (1-2) using the 'expm' function of Matlab for different methods: (A) using an efficient implementation of (1-2) as in [4], (B) using our approach (4-7), (C) using (8-9). All the computations were carried out with Matlab 7.4.0.

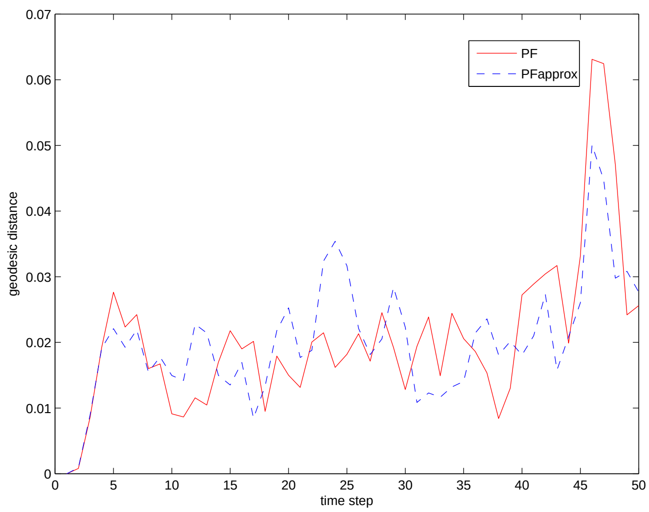

Fig. 1. Geodesic distance between the data and the filtered data returned by the PF and PFapprox methods with $\sigma_{\text{data}} = 0.05$, 1000 particles for $\|A_0\|_F = 0.2236$.

small with respect to n. When p increases and gets close to $\frac{n}{2}$, the approximation by retractions and vector transports (8-9) becomes interesting. A stochastic piecewise geodesic trajectory on $\mathrm{G}(4,2)$ with $\sigma_{\text{model}} = 0.05\|A_0\|_F$ was also simulated according to (1-2). The particle filtering technique was applied on this data set and the geodesic distance between the data and the filtered data was computed as a measure of error. This was carried out for the particle filtering technique using the geodesic process (4-7) denoted by PF and using our approximation (8-9) with only a QR decomposition denoted by PFapprox. If the norm of A_0 is small, the subspaces vary slowly and the level of the error using the PF or PFapprox methods are equivalent, see Fig.1. But when the norm of A_0 is increased, the subspaces move more rapidly and the time-averaged error of the PFapprox method becomes bigger than for the PF method as shown in Table 3.

$\|A_0\|_F$	mean error PF	mean error PFapprox
0.1118	0.0186	0.0190
0.2236	0.0231	0.0250
0.3354	0.0269	0.0358
0.4472	0.0342	0.1147

Table 3. Time-average errors over 50 time steps averaged over 50 different realizations of a stochastic piecewise geodesic trajectory using the same initial state. All the experiments were carried out with $\sigma_{\text{data}} = 0.05$ and 1000 particles.

6. CONCLUSIONS

We have presented an efficient way to implement the particle filtering technique of [1] that reduces the spatial complexity per particle from $n^2 + (n-p)p$ to $2np$ and the computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(np^2)$. An approximation of the geodesic process has been also presented to avoid the computation of the SVD. This approximation is particularly interesting when p gets close to $\frac{n}{2}$ and when the subspaces vary slowly.

7. REFERENCES

[1] Anuj Srivastava and Eric Klassen, "Bayesian and geometric subspace tracking," *Adv. in Appl. Probab.*, vol. 36, no. 1, pp. 43–56, 2004.

[2] Tiesheng Wang, Andrew Backhouse, and Irene Y.H. Gu, "Online subspace learning in Grassmann manifold for moving object tracking in video," in *Proceedings of IEEE international conf. Acoustics, Speech, and Signal Processing (ICASSP'08)*, 2008, pp. 969–972.

[3] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, NJ, 2008.

[4] Kyle Gallivan, Anuj Srivastava, Xiuwen Liu, and Paul Van Dooren, "Efficient algorithms for inferences on Grassmann manifolds," in *Proceedings of 12 th IEEE Workshop on Statistical Signal Processing*, 2003, pp. 315–318.

[5] Alan Edelman, Tomás A. Arias, and Steven T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353 (electronic), 1999.