Particle Flow Particle Filter using Gromov's method

Conference Paper · December 2019

DOI: 10.1109/CAMSAP45676.2019.9022494

2 authors, including:

Soumyasundar Pal
McGill University

Particle Flow Particle Filter using Gromov's method

Soumyasundar Pal and Mark Coates
Department of Electrical and Computer Engineering
McGill University, 3480 University Street, Montreal, Quebec, Canada H3A2A7
Email: soumyasundar.pal@mail.mcgill.ca, mark.coates@mcgill.ca

***Abstract*—Particle flow filters obtain impressive results in challenging high dimensional, non-linear sequential state estimation problems. In contrast to a particle filter, which uses importance sampling to approximate the posterior distribution of the state, the flow based algorithms solve a differential equation to migrate the particles from the prior to the posterior distribution. However, the particles after the flow are not true samples of the posterior distribution due to strong model assumptions required for the derivation of the flow and the approximations associated with the numerical solution. This affects performance adversely in many highly non-linear, non-Gaussian filtering problems. Particle Flow Particle Filters (PFPF) adapt the particle flow procedure to construct a proposal density inside the particle filter. These techniques can outperform the underlying particle flow algorithms by compensating for the approximations in the flow calculations via update of importance weights after the flow, at the cost of a negligible increase in the computational complexity. Most of the PFPF approaches have focused on using a deterministic particle flow. In this paper, we develop a PFPF algorithm using a stochastic particle flow based on Gromov's method. Numerical simulations are conducted to examine when the proposed method offers advantages compared to existing techniques.**

***Index Terms*—** non-linear sequential state estimation, particle flow, particle filter, high-dimensional filtering.

I. Introduction

Particle filters [1] have become a standard tool for performing sequential state estimation in a Bayesian framework. They employ sequential importance sampling to approximate the posterior distribution of the states in a recursive manner. However, if the measurements are highly informative or the state dimension is high, most of the particles fall in regions of state space where the likelihood is very low after a few time steps. This leads to negligible weights for the majority of particles which results in a deterioration in the approximation of the posterior [2]–[6].

Although there have been multiple proposals [5]–[10] to address the weight degeneracy issue, the applicability of the methods are somewhat limited, either because they impose strong assumptions on the structure of the posterior or involve a very high computational cost.

Particle flow algorithms [11]–[15] can achieve impressive performance for a reduced computational overhead in many challenging filtering tasks. In these filters, particles are gradually migrated from the prior to the posterior distribution by solving differential equations. The idealized modelling assumptions and the numerical approximations needed to implement these filters can lead to particles not being distributed according to the posterior. To address this, several recent algorithms combine particle flow and particle filtering [15]–[19]. In particular, [19] introduces the particle flow particle filter (PFPF), which uses particle flow to construct the proposal distribution in a particle filtering framework. As a result, the convergence results for any particle filter still apply, while the flow constructs an efficient proposal distribution.

The PFPF of [19] relies on a zero-diffusion particle flow [13], [20]. Recent work [21], [22] has shown that stochastic algorithms can significantly outperform the deterministic particle flow algorithms. In this paper, we propose to incorporate a variant of stochastic particle flow, called the Gromov flow in the particle filtering framework.

The paper is organized as follows. Section II states the filtering task that we address. Section III reviews the geodesic particle flow and the Gromov flow filters [22], [23]. Section IV introduces the proposed particle flow particle filter using Gromov flow and Section V presents and discusses the results of numerical simulation experiments.

II. Problem Statement

We address the task of sequential state estimation in discrete time for a hidden Markov model. The unobserved state and the observation at time k are denoted by $x_k \in \mathbb{R}^d$ and $z_k \in \mathbb{R}^S$ respectively. We assume that the hidden state x_k evolves according to a first order Markov process, i.e., it is independent of all states before time $k-1$ conditioned on the state x_{k-1}. The measurement z_k is assumed to be independent of all past measurements and past states conditioned on the current state x_k. The state evolution and measurements are described by the following model:

$$x_0 \sim p(x_0)\,, \tag{1}$$

$$x_k = g_k(x_{k-1}, v_k) \quad \text{for } k \geq 1\,, \tag{2}$$

$$z_k = h_k(x_k, w_k) \quad \text{for } k \geq 1\,. \tag{3}$$

Here $p(x_0)$ is an initial probability density function of state x_0, $g_k : \mathbb{R}^d \times \mathbb{R}^{d'} \to \mathbb{R}^d$ models the dynamics of the unobserved state x_k and the measurement model $h_k : \mathbb{R}^d \times \mathbb{R}^{S'} \to \mathbb{R}^S$ specifies the dependence of the observation z_k on state x_k. The process and measurement noises are denoted by $v_k \in \mathbb{R}^d$ and $w_k \in \mathbb{R}^{S'}$ respectively.

Given a set of observations $z_{1:k} = \{z_1, \ldots, z_k\}$ and an initial distribution $p(x_0)$, our goal is to track the marginal posterior distribution $p(x_k|z_{1:k})$. This can be used to form estimates of x_k and to assess uncertainty.

III. Geodesic Particle Flow and Gromov's method

Suppose that we have a set of N_p particles $\{x^i_{k-1}\}_{i=1}^{N_p}$, which are distributed according to the posterior distribution at time $k-1$. After propagating these particles through the dynamic model, we obtain particles $\{\tilde{x}_k^i\}_{i=1}^{N_p}$ approximating the predictive posterior distribution at time k. Particle flow methods then solve a stochastic differential equation to migrate the particles so that they become samples of the posterior distribution at time k.

We consider the particle flow as a background stochastic process η_λ in a pseudo time interval $\lambda \in [0,1]$. Since the particle flow only migrates the particles within a single time step, we omit the time index k temporarily to simplify the notation. η_λ^i is used to denote the i-th realization of the stochastic process. The flow is initialized at $\eta_0^i = \tilde{x}_k^i$, for $i = 1, 2, \ldots, N_p$.

We assume the particle flow follows a stochastic differential equation of the following form

$$d\eta_\lambda^i = f(\eta_\lambda^i, \lambda)d\lambda + d\nu^i \,, \tag{4}$$

where $f(\eta_\lambda^i, \lambda)$ is referred to as the flow and $\nu^i(\lambda)$ is called the diffusion term. The covariance matrix of the stochastic process $\nu^i(\lambda)$ is denoted $Q(\eta_\lambda^i, \lambda)$, i.e., $d\nu^i d{\nu^i}^T = Q(\eta_\lambda^i, \lambda)d\lambda$. The covariance matrix is required to be positive semi-definite, i.e., $Q(\eta_\lambda^i, \lambda) \succcurlyeq \mathbf{0}$. The allowable choices of f and Q are governed by the Fokker-Planck equation [24]. By imposing different constraints on the flow, we can obtain a variety of particle flow filters. For example, setting $Q(\eta_\lambda^i, \lambda) = \mathbf{0}$ leads to the *zero diffusion* particle flow filters [12], [13], [24]–[27]. Although very few particle flow algorithms are analytically tractable, calculation of the flow becomes analytically tractable when the predictive posterior and the likelihood distributions are both Gaussian and the measurement model is linear, i.e., $\eta_0^i \sim N(\bar{\eta}_0, \bar{P}), z = H\eta_\lambda^i + w \sim \mathcal{N}(H\eta_\lambda^i, R)$. The predictive covariance $\bar{P}$ and the measurement covariance R are both positive definite, and H is the measurement matrix.

A. The Geodesic Flow

Using the linearized approximation of the measurement model at the particle locations, the flow parameters for the geodesic flow [23] are computed as follows:

$$f(\eta_\lambda^i, \lambda) = A^i(\lambda)\eta_\lambda^i + b^i(\lambda) \,, \tag{5}$$

$$Q(\eta_\lambda^i, \lambda) = \mathbf{0} \,, \tag{6}$$

where

$$\begin{aligned} M^i(\lambda) &= \big(\bar{P}^{-1} + \lambda H^i(\lambda)^T R^{-1} H^i(\lambda)\big)^{-1} \,, \\ A^i(\lambda) &= -M^{(i)}(\lambda) H^i(\lambda)^T R^{-1} H^i(\lambda) \,, \\ b^i(\lambda) &= M^{(i)} H^i(\lambda)^T R^{-1} \big(z - e^i(\lambda)\big) \,. \end{aligned} \tag{7}$$

Here $H^i(\lambda) = \left.\dfrac{\partial h(\eta)}{\partial \eta}\right|_{\eta=\eta_\lambda^i}$ and $e^i(\lambda) = h(\eta_\lambda^i) - H^i(\lambda)\eta_\lambda^i$.

B. The Gromov Flow

In the Gromov flow [22], the expression for the flow f is exactly the same as in eq. (5). However, this flow is stochastic, i.e., it has a non-zero diffusion covariance as follows:

$$\begin{aligned} Q(\eta_\lambda^i, \lambda) = \big(\bar{P}^{-1} + \lambda H^i(\lambda)^T R^{-1} H^i(\lambda)\big)^{-1} \times \\ H^i(\lambda)^T R^{-1} H^i(\lambda) \big(\bar{P}^{-1} + \lambda H^i(\lambda)^T R^{-1} H^i(\lambda)\big)^{-1} \,. \end{aligned} \tag{8}$$

C. Numerical Implementation

We employ discretized pseudo-time integration to compute the approximate solution of the flow equation (4). We identify a sequence of discrete steps at N_λ positions, $[\lambda_1, \lambda_2, \ldots, \lambda_{N_\lambda}]$, where $0 = \lambda_0 < \lambda_1 < \ldots < \lambda_{N_\lambda} = 1$. The step size $\epsilon_j = \lambda_j - \lambda_{j-1}$ for $j = 1, \ldots, N_\lambda$ can vary but we require that $\sum_{j=1}^{N_\lambda} \epsilon_j = \lambda_{N_\lambda} - \lambda_0 = 1$. For computation of (7) and (8) we need $H^i(\lambda_j)$, which is obtained by computing the derivative of the measurement function h at $\eta_{\lambda_{j-1}}^i$. The update for the geodesic flow becomes

$$\eta_{\lambda_j}^i = \eta_{\lambda_{j-1}}^i + \epsilon_j\big(A^i(\lambda_j)\eta_{\lambda_{j-1}}^i + b^i(\lambda_j)\big) \,. \tag{9}$$

Similarly, the update for the Gromov flow is given as

$$\eta_{\lambda_j}^i = \eta_{\lambda_{j-1}}^i + \epsilon_j\big(A^i(\lambda_j)\eta_{\lambda_{j-1}}^i + b^i(\lambda_j)\big) + \sqrt{\epsilon_j}\nu^i(\lambda_j) \tag{10}$$

where $\nu^i(\lambda_j) \sim \mathcal{N}\big(\mathbf{0}, Q(\eta_{\lambda_j}^i, \lambda_j)\big)$.

IV. Particle Flow Particle Filter (Gromov)

Assuming we have a set of weighted particles $\{\omega_{k-1}^i, x_{k-1}^i\}_{i=1}^{N_p}$ approximating the posterior at time $k-1$, we generate two sets of particles $\{\bar{\eta}_0^i, \eta_0^i\}_{i=1}^{N_p}$ as follows:

$$\zeta_k = \mathbb{E}[v_k] \,, \tag{11}$$

$$\bar{\eta}_0^i = g_k(x_{k-1}^i, \zeta_k) \,, \tag{12}$$

$$V^i = \mathbf{Cov}[x_k | x_{k-1}^i] \,, \tag{13}$$

$$\eta_0^i \sim \mathcal{N}\big(\bar{\eta}_0^i, V^i\big) \,. \tag{14}$$

If the process noise is additive Gaussian with a covariance matrix which does not depend on the previous state x_{k-1}, then η_0^i can equivalently be sampled by propagating the particle x_{k-1}^i through the dynamic model. However, for non-Gaussian dynamic models, a pointwise Gaussian approximation is required for sampling η_0^i. The distribution of η_0^i conditioned on x_{k-1}^i is Gaussian. We construct an auxiliary geodesic flow starting from $\bar{\eta}_0^i$ to compute the flow and diffusion parameters and use the same deterministic parameters to perform the Gromov flow starting at η_0^i. We note that this procedure ensures that the distribution of $\eta_{\lambda_j}^i$ conditioned on (x_{k-1}^i, z_k) remains Gaussian. We define

$$\mu_{\lambda_j}^i = \mathbb{E}[\eta_{\lambda_j}^i | x_{k-1}^i, z_k] \,, \tag{15}$$

$$\Sigma_{\lambda_j}^i = \mathbf{Cov}[\eta_{\lambda_j}^i | x_{k-1}^i, z_k] \,. \tag{16}$$

with initializations $\mu_0^i = \bar{\eta}_0^i$ and $\Sigma_0^i = V^i$. Then the update of these parameters can be computed using the following recursion:

$$\mu_{\lambda_j}^i = \big(\mathbf{I} + \epsilon_j A(\lambda_j)\big)\mu_{\lambda_{j-1}}^i + \epsilon_j b(\lambda_j) \,, \tag{17}$$

$$\Sigma_{\lambda_j}^i = \big(\mathbf{I} + \epsilon_j A(\lambda_j)\big)\Sigma_{\lambda_{j-1}}^i \big(\mathbf{I} + \epsilon_j A(\lambda_j)\big)^T + \epsilon_j Q(\bar{\eta}_{\lambda_j}^i, \lambda_j) \,, \tag{18}$$

After the flow is complete, we set $x_k^i = \eta_1^i$ and the proposal distribution becomes $q(\eta_1^i | x_{k-1}^i, z_k) = \mathcal{N}\big(\eta_1^i; \mu_1^i, \Sigma_1^i\big)$, which can be evaluated in closed form. The algorithm is summarized in Algorithm 1.

Algorithm 1 Particle Flow Particle Filter (Gromov)

Initialization: Draw particles $\{x_0^i\}_{i=1}^{N_p}$ from the prior $p(x_0)$. Set $\{\omega_0^i\}_{i=1}^{N_p} = \dfrac{1}{N_p}$, compute $\hat{x}_0 = \sum_{i=1}^{N_p} \omega_0^i x_0^i$.
for $k = 1$ to K **do**
  for $i = 1, \dots, N_p$ **do**
    Apply EKF/UKF prediction to compute $\bar{P}_k^i$: $\{x_{k-1}^i, P_{k-1}^i\} \rightarrow \{m_{k|k-1}^i, \bar{P}_k^i\}$
    Calculate $\bar{\eta}_0^i = g_k(x_{k-1}^i, \zeta_k)$
    Calculate $V^i = \mathbf{Cov}[x_k | x_{k-1}^i]$
    Sample $\eta_0^i \sim \mathcal{N}(\bar{\eta}_0^i, V^i)$
    Initialize $\mu_0^i = \bar{\eta}_0^i, \Sigma_0^i = V^i$ and $\lambda_0 = 0$.
    for $j = 1, \dots, N_\lambda$ **do**
      Set $\lambda_j = \lambda_{j-1} + \epsilon_j$
      Calculate $A^i(\lambda_j)$ and $b^i(\lambda_j)$ from eq. (7) and $Q(\bar{\eta}_{\lambda_j}^i, \lambda_j)$ from eq. (8) with linearization performed at $\bar{\eta}_{\lambda_{j-1}}^i$ and with $\bar{P} = \bar{P}_k^i$
      Migrate auxiliary particle using geodesic flow: $\bar{\eta}_{\lambda_j}^i = \bar{\eta}_{\lambda_{j-1}}^i + \epsilon_p (A^i(\lambda_j) \bar{\eta}_{\lambda_{j-1}}^i + b^i(\lambda_j))$
      Migrate particle using Gromov flow: $\eta_{\lambda_j}^i = \eta_{\lambda_{j-1}}^i + \epsilon_p (A^i(\lambda_j) \eta_{\lambda_{j-1}}^i + b^i(\lambda_j)) + \sqrt{\epsilon_j} \nu^i(\lambda_j)$, where $\nu^i(\lambda_j) \sim \mathcal{N}(\mathbf{0}, Q(\bar{\eta}_{\lambda_j}^i, \lambda_j))$
      Compute $\mu_{\lambda_j}^i$ and $\Sigma_{\lambda_j}^i$ from eq. (17) and (18)
    end for
    Set $x_k^i = \eta_1^i$
    Calculate importance weights:

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(x_k^i | x_{k-1}^i) p(z_k | x_k^i)}{\mathcal{N}(x_k^i; \mu_1^i, \Sigma_1^i)}$$

  end for
  for $i = 1, \dots, N_p$ **do**
    Normalize $\omega_k^i = \omega_k^i / \sum_{s=1}^{N_p} \omega_k^s$
    Apply EKF/UKF update to compute P_k^i: $\{m_{k|k-1}^i, \bar{P}_k^i\} \rightarrow \{m_{k|k}^i, P_k^i\}$
  end for
  Estimate $\hat{x}_k = \sum_{i=1}^{N_p} \omega_k^i x_k^i$
  (Optional) resample particles : $\{x_k^i, P_k^i, \omega_k^i\}_{i=1}^{N_p}$ to obtain $\{x_k^i, P_k^i, \dfrac{1}{N_p}\}_{i=1}^{N_p}$
end for

V. Numerical Experiments and Results

We conduct numerical simulations for two scenarios. The first is a linear Gaussian setup for state estimation in a large spatial sensor network, which allows us to compare the performance of the proposed filter with the optimal Kalman filter. The second is a multi-target acoustic tracking problem with a highly non-linear measurement model that requires the use of sophisticated particle filters to obtain accurate state estimates. We compare the proposed PFPF (Gromov) algorithm with the PFPF algorithms [19], the Bootstrap Particle Filter (BPF) [1] and various particle flow [13], [20], [22], [23] techniques. In addition, we also implement the Gaussian Particle Flow Importance Sampling (GPFIS) algorithm [17] for the non-linear example. All numerical simulations are executed using Matlab on an Intel i7-4770K, 3.50GHz CPU and 32GB RAM.

A. Linear Gaussian example

We use the spatial sensor network setup proposed in [6]. The network consists of d sensors placed uniformly on a two-dimensional grid $\{1, 2, ..., \sqrt{d}\} \times \{1, 2, ..., \sqrt{d}\}$ and we set $d = 64$ in this example. At each time step k, each sensor i records a measurement z_k^i, independently of the other sensors, about the underlying state x_k^i at its physical location. The evolution of the state vector $x_k = [x_k^1, x_k^2, ..., x_k^d]^T \in \mathbb{R}^d$ in time follows a linear model with Gaussian noise and the measurement vector $z_k = [z_k^1, z_k^2, ..., z_k^d]^T \in \mathbb{R}^d$ is also linear in state x_k.

The dynamic model and the measurement model are:

$$x_k = \alpha x_{k-1} + v_k \,, \tag{19}$$

$$z_k = x_k + w_k \,, \tag{20}$$

where $\alpha = 0.9$, the process noise $v_k \sim \mathcal{N}(\mathbf{0}, \Sigma)$. The (i,j)-th entry of Σ is given as $\Sigma_{i,j} = \alpha_0 e^{-\frac{\|R^i - R^j\|_2^2}{\beta}} + \alpha_1 \delta_{i,j}$, where $R^i \in \mathbb{R}^2$ denotes the physical location of the i-th sensor on the grid and $\delta_{i,j}$ is the Kronecker delta symbol ($\delta_{i,i} = 1$ and $\delta_{i,j} = 0$ for $i \neq j$). The structure of Σ implies that the correlation between sensors decreases with the increase of the Euclidean distance between them. We set $\alpha_0 = 3$, $\alpha_1 = 0.01$ and $\beta = 20$, as in [6]. The measurement noise is denoted by $w_k \sim \mathcal{N}(\mathbf{0}, \sigma_w^2 \mathbf{I})$. We set $\sigma_w = 0.5$. The true states start at $x_0 = \mathbf{0}$. The experiment is repeated for 100 Monte Carlo trials for $T = 10$ time steps.

TABLE I: Average MSE, $\text{MSE}_{\log Z}^{(rel)}$, ESS (if applicable) and execution time per step in the linear Gaussian example.

Algorithm	No. particles	Avg. MSE	Avg. $\text{MSE}_{\log Z}^{(rel)}$	Avg. ESS	Exec. time (s)
KF	-	0.07	0	-	0.002
PFPF (Gromov)	200	0.09	0.0004	30.6	2.3
PFPF (Geodesic)	200	0.09	0.0340	3.5	2.1
PFPF (LEDH)	200	0.09	0.0005	25.1	1.9
PFPF (EDH)	200	0.09	0.0006	21.7	0.015
PFPF (EDH)	10^4	0.08	0.0001	852	0.2
Gromov Flow	200	0.07	-	-	2.2
Geodesic Flow	200	0.07	-	-	1.8
LEDH	200	0.07	-	-	1.4
EDH	200	0.07	-	-	0.011
BPF	200	1.10	2.813	1.04	0.001
BPF	10^6	0.20	0.0265	1.62	2.5

In this linear Gaussian example, the true posterior distribution of the states is obtained from a Kalman filter, which also allows us to compute the normalizing constant of the posterior distribution in a closed form. In addition to reporting the average Mean Square Error (MSE) in the state estimates, we also compute the relative MSE for the log of the normalizing constants $\text{MSE}_{\log Z}^{(rel)}$ [28] from the particle filters. From Table I we observe that the particle flow algorithms obtain similar average MSE as the Kalman filter, since the linear Gaussian models match their model assumptions. The proposed PFPF (Gromov) algorithm performs comparably to the other PFPF algorithms in terms of state estimation. Moreover, it achieves the highest average effective sample size (ESS) [29] and the lowest average $\mathrm{MSE}^{(rel)}_{\log Z}$ among all the PFPF algorithms employing the same number of particles. In particular, comparison with PFPF (Geodesic) indicates that the inclusion of the diffusion term in the particle flow equation diversifies the particles and in turn results in a more efficient proposal distribution. However, a much higher ESS can be obtained from the PFPF (EDH) algorithm by increasing the number of particles for this example, with negligible computational overhead.

B. Non-linear example

We consider a multi-target tracking problem for superpositional sensors with low measurement noise. The scenario is similar to those examined in [19], [30]. There are $C = 4$ targets moving independently according to a constant velocity model in a region of size 40m $\times$ 40m, equipped with 25 superpositional sensors deployed uniformly. The dynamic model for the c-th target is specified as $x_k^{(c)} = F x_{k-1}^{(c)} + v_k^{(c)}$, where $x_k^{(c)} = [\mathrm{x}_k^{(c)}, \mathrm{y}_k^{(c)}, \dot{\mathrm{x}}_k^{(c)}, \dot{\mathrm{y}}_k^{(c)}]^T$ is comprised of the position coordinates and velocities. F denotes the state transition matrix for each target. The process noise $v_k^{(c)} \sim \mathcal{N}(\mathbf{0}, V)$. We set

$$F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } V = \frac{1}{20} \begin{bmatrix} \frac{1}{3} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{bmatrix}.$$

Each target emits sound of amplitude $\Psi = 10$ and the sum of the attenuated amplitudes from all targets is recorded at each sensor. The measurement function for the s-th sensor at R^s is:

$$\bar{z}^s(x_k) = \sum_{c=1}^{C} \frac{\Psi}{\|(\mathrm{x}_k^{(c)}, \mathrm{y}_k^{(c)})^T - R^s\|_2 + d_0}. \tag{21}$$

We set $d_0 = 0.1$. The measurements are corrupted with Gaussian noise, i.e., the noisy observation z_k^{c} from s-th sensor is a realization of $\mathcal{N}\big(\bar{z}^s(x_k), \sigma_w^2\big)$. We simulate 100 random state trajectories and corresponding measurement sequences for 40 time steps. Each algorithm is run 5 times on each trajectory starting from different initial distributions. We also assume that the dynamic model is not known accurately during tracking. We therefore use an inflated covariance matrix for the process noise for all the filters (we copy the approach from [19] — please see this paper for details regarding the inflation and the initialization of the state distribution). We consider two different noise values: $\sigma_w = 0.1$ and 0.001. The average optimal sub-pattern assignment (OSPA) metric [31] is used as the performance measure.

From Table II, we observe that the proposed PFPF (Gromov) algorithm attains the smallest average OSPA metric for both values of σ_w. However, the relative improvement is more prominent for extremely informative measurements ($\sigma_w = 0.001$). We also note that the proposed PFPF (Gromov) algorithm achieves the largest average ESS among all the competing particle filters with same number of particles. Although the PFPF (Geodesic) algorithm achieves similar tracking performance as the PFPF (Gromov) algorithm, it suffers from weight degeneracy. The PFPF (LEDH) algorithm with 500 particles offers much better performance in comparison to the PFPF (EDH) algorithm with 10^5 particles. This indicates that for highly non-linear measurement models, performing local linearization for computing the particle flow becomes crucial. The EKF, the UKF and the particle flow algorithms have large error, probably because of the strong non-linearity of the measurement function. The comparison between the particle flow algorithms and the corresponding PFPF algorithms demonstrates the improvement in performance due to the importance sampling step. The GPFIS algorithm uses approximate Gaussian flows to sample from non-Gaussian posterior distributions, followed by a computationally expensive weight update step. However, while it offers improved tracking performance compared to the particle flow algorithms (at much higher computational cost), its performance is significantly worse than the PFPF algorithms except for PFPF (EDH).

TABLE II: Average OSPA metric, average ESS (if applicable) and execution time per step for the multi-target acoustic tracking example.

Algorithm	No. particles	$\sigma_w = 0.1$		$\sigma_w = 0.001$		Exec. time (s)
		Avg. OSPA (m)	Avg. ESS	Avg. OSPA (m)	Avg. ESS	
PFPF(Gromov)	500	0.78	72.3	0.24	215.1	1.3
PFPF(Geodesic)	500	0.78	7.6	0.25	6.09	1.15
PFPF(LEDH)	500	0.79	45	0.28	1.02	0.9
PFPF(EDH)	10^5	2.42	1680	2.39	1.1	0.5
GPFIS	500	0.93	30	1.01	29.7	66.8
EKF	-	5.74	-	14.4	-	0.00003
UKF	-	4.91	-	37.9	-	0.005
Gromov Flow	500	1.91	-	2.41	-	1.25
Geodesic Flow	500	2.00	-	2.45	-	0.9
LEDH	500	2.19	-	1.23	-	0.8
EDH	10^5	2.79	-	2.45	-	0.45
BPF	10^6	1.10	6.3	1.14	1.0	3.0

VI. Conclusion

We developed a particle flow particle filter that uses a stochastic particle flow based on Gromov's method. Experimental results suggest that the non-zero diffusion term helps in diversifying particles in challenging filtering problems with high dimensionality of the state vector and/or very informative measurements. Although the proposed filter achieves the highest average ESS among the competing PFPF algorithms with the same number of particles in all our experiments, this translates to a relatively small improvement in the accuracy of the state estimates in the multi-target tracking example. Future research will investigate ways to reduce the computational overhead by considering an EDH type flow and assess performance via a more extensive experimental evaluation.

REFERENCES

[1] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc. F Radar and Signal Process.*, vol. 140, no. 2, Apr. 1993, pp. 107–113.

[2] P. Bickel, B. Li, and T. Bengtsson, "Sharp failure rates for the bootstrap particle filter in high dimensions," in *Pushing the limits of contemporary statistics: Contributions in honor of Jayanta K. Ghosh*. Institute of Mathematical Statist., May 2008, pp. 318–329.

[3] T. Bengtsson, P. Bickel, and B. Li, "Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems," in *Probability and Statistics: Essays in Honor of David A. Freedman*. Beachwood, OH, USA: Institute of Mathematical Statistics, Apr. 2008, vol. 2, pp. 316–334.

[4] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Mon. Weather Rev.*, vol. 136, no. 12, pp. 4629–4640, Dec. 2008.

[5] A. Beskos, D. Crisan, and A. Jasra, "On the stability of sequential Monte Carlo methods in high dimensions," *Ann. Appl. Probab.*, vol. 24, no. 4, pp. 1396–1445, Aug. 2014.

[6] F. Septier and G. W. Peters, "Langevin and Hamiltonian based sequential MCMC for efficient Bayesian filtering in high-dimensional spaces," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 2, pp. 312–327, Mar. 2016.

[7] P. M. Djurić, T. Lu, and M. F. Bugallo, "Multiple particle filtering," in *Proc. IEEE Intl. Conf. Acoustics, Speech and Signal Process.*, vol. 3, Honolulu, HI, USA, Apr. 2007, pp. 1181–1184.

[8] P. M. Djurić and M. F. Bugallo, "Particle filtering for high-dimensional systems," in *Proc. IEEE Intl. Workshop Comput. Advances in Multi-Sensor Adaptive Process.*, St. Martin, France, Dec. 2013, pp. 352–355.

[9] A. Beskos, D. Crisan, A. Jasra, K. Kamatani, and Y. Zhou, "A stable particle filter in high-dimensions," *arXiv:1412.3501*, Dec. 2014.

[10] P. Rebeschini and R. van Handel, "Can local particle filters beat the curse of dimensionality?" *Ann. Appl. Probab.*, vol. 25, no. 5, pp. 2809–2866, Oct. 2015.

[11] U. D. Hanebeck, K. Briechle, and A. Rauh, "Progressive Bayes: a new framework for nonlinear state estimation," in *Proc. SPIE Multisensor, Multisource Info. Fusion: Architectures, Algorithms, and Applications*, vol. 5099, Orlando, FL, USA, Apr. 2003, pp. 256–267.

[12] F. Daum and J. Huang, "Nonlinear filters with log-homotopy," in *Proc. SPIE Signal and Data Process. Small Targets*, San Diego, CA, USA, Sep. 2007, p. 669918.

[13] F. Daum, J. Huang, and A. Noushin, "Exact particle flow for nonlinear filters," in *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, Orlando, FL, USA, Apr. 2010, p. 769704.

[14] M. A. Khan and M. Ulmke, "Non-linear and non-Gaussian state estimation using log-homotopy based particle flow filters," in *Proc. Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, Bonn, Germany, Oct. 2014, pp. 1–6.

[15] F. E. de Melo, S. Maskell, M. Fasiolo, and F. Daum, "Stochastic particle flow for nonlinear high-dimensional filtering problems," *arXiv:1511.01448*, Nov. 2015.

[16] S. Reich, "A guided sequential Monte Carlo method for the assimilation of data into stochastic dynamical systems," in *Recent Trends in Dynamical Systems*. Basel, Switzerland: Springer, 2013, vol. 35, pp. 205–220.

[17] P. Bunch and S. Godsill, "Approximations of the optimal importance density using gaussian particle flow importance sampling," *J. Amer. Statist. Assoc.*, vol. 111, no. 514, pp. 748–762, Aug. 2016.

[18] J. Heng, A. Doucet, and Y. Pokern, "Gibbs flow for approximate transport with applications to Bayesian computation," *arXiv:1509.08787*, Sep. 2015.

[19] Y. Li and M. Coates, "Particle filtering with invertible particle flow," *IEEE Trans. Signal Process.*, vol. 65, no. 15, pp. 4102–4116, Aug. 2017.

[20] T. Ding and M. J. Coates, "Implementation of the daum-huang exact-flow particle filter," in *Proc. IEEE Statist. Signal Process. Workshop (SSP)*, Ann Arbor, MI, USA, Aug. 2012, pp. 257–260.

[21] F. Daum, J. Huang, and A. Noushin, "Generalized Gromov method for stochastic particle flow filters," in *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, Anaheim, CA, USA, May 2017, p. 102000I.

[22] F. Daum, J. Huang, and A. Noushin, "New theory and numerical results for Gromov's method for stochastic particle flow filters," in *Proc. Int. Conf. Info. Fusion*, Cambridge, UK, July 2018, pp. 108–115.

[23] F. Daum and J. Huang, "Particle flow with non-zero diffusion for nonlinear filters," in *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, Baltimore, MD, USA, May 2013, p. 87450P.

[24] ——, "Exact particle flow for nonlinear filters: seventeen dubious solutions to a first order linear underdetermined PDE," in *Proc. Asilomar Conf. Signals, Systems and Computers*, Pacific Grove, CA, USA, Nov. 2010, pp. 64–71.

[25] ——, "Particle flow for nonlinear filters with log-homotopy," in *Proc. SPIE Signal and Data Process. Small Targets*, Orlando, FL, USA, Apr. 2008, p. 696918.

[26] F. Daum, J. Huang, A. Noushin, and M. Krichman, "Gradient estimation for particle flow induced by log-homotopy for nonlinear filters," in *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, Orlando, FL, USA, Apr. 2009, p. 733602.

[27] F. Daum, J. Huang, and A. Noushin, "Coulomb's law particle flow for nonlinear filters," in *Proc. SPIE Conf. Signal Process., Sensor Fusion, Target Recog.*, San Diego, CA, USA, Sep. 2011, p. 81370B.

[28] Y. Li, S. Pal, and M. J. Coates, "Invertible particle flow-based sequential MCMC with extension to Gaussian mixture noise models," *IEEE Trans. Signal Process.*, vol. 67, no. 9, pp. 2499–2512, May 2019.

[29] A. Kong, "A note on importance sampling using standardized weights," Univ. Chicago, Dept. Statist., Tech. Rep. 348, 1992.

[30] O. Hlinka, O. Sluciak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Distributed Gaussian particle filtering using likelihood consensus," in *Proc. IEEE Intl. Conf. Acoust., Speech and Signal Process.*, Prague, Czech Republic, May 2011, pp. 3756–3759.

[31] D. Schuhmacher, B. T. Vo, and B. N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug 2008.