Variational Bayesian Point Set Registration

Xiaoyue Jiang[‡*], Hang Yu[†], Michael Hoy[*], Justin Dauwels[*†]
[*] ST Engineering – NTU Corporate Lab, [†]Centre for System Intelligence and Efficiency (EXQUISITUS)
School of Electrical and Electronic Engineering, Nanyang Technological University Singapore

***Abstract*—Point set registration presents unique significance in Lidar-based intelligent vehicle localization and mapping. It involves registering point sets of the same scene observed from different positions by determining their relative spatial transformation. However, due to the noise and outliers in the point sets and initial misalignment, existing methods suffer from the issues of low accuracy or large computational cost. In this paper, we propose a novel Bayesian state space model to describe the sequential point registration problem. Specifically, we specify the transformations to be the latent states and further assume that they vary smoothly across time. The point clouds are represented as Gaussian mixture models that change accordingly with the transformation. We develop a stochastic variational Bayesian inference algorithm to learning the distributions of the transformation, which automatically strikes a balance between mapping every two consecutive point clouds and the temporal smoothness of the transformation. Experimental results based synthetic data show that the proposed variational Bayesian point set registration (VB-PSR) algorithm achieves higher accuracy with comparable or less time and resources, in comparison with the state-of-the-art methods.**

I. Introduction

In autonomous driving, a robot needs to locate itself while simultaneously building the environmental map through sensors' perception. This simultaneous localization and mapping (SLAM) process generally consists of two parts: the front end and the back end. The front end performs frame-by-frame pose estimation, while the back end refines this initial solution by correlating observations. Therefore, accurate odometry resulting from the front end is crucial for the vehicle to work smoothly and efficiently. Compared with visual odometry (VO), Lidar odometry (LO) is of higher accuracy and better stability, especially in the outdoor environment. The acquisition of relative pose between every two consecutive Lidar point clouds is referred to as point set registration.

According to the transformation type, point set registration can be classified as rigid and nonrigid. Given two point sets, the rigid registration only consists of translation and rotation, whereas the nonrigid typically involves nonlinear transformation. In the context of SLAM applications, here we focus on the rigid registration problem.

The substantial literature on rigid point registration can be classified into three categories: distance-optimization-based, probability-based, and filtering-based methods.

The distance-optimization-based algorithms estimate the transformation by finding the correspondence between the points in every two point clouds (i.e., frames or scans). Among all algorithms in this group, the iterative closest point (ICP) algorithm [1] stands out due to its simplicity and efficiency. According to an estimated transformation, ICP maps the first point cloud onto the second one and establish the correspondence between the points from the two clouds by associating every point in the first cloud with the one in the second cloud that is closest in the distance. The transformation is further updated such that the distance between the pairs of points with correspondence is minimized. These two steps are then iterated until convergence. The computational complexity of ICP is only $\mathcal{O}(MD\log(MD))$, where M is the number of points in the template point set, and D is the dimension of the point space. However, the point matching process in ICP is susceptible to the initial guess of the transformation and the noise and outliers that often exist in the point frame.

As a remedy to the aforementioned problems, the second group of methods, namely probability-based methods, abandon the point-wise registration in the distance-optimization-based algorithms, and instead represent point sets by probability distributions. One popular approach in this group is the normal distributions transform (NDT) algorithm [2]. In this algorithm, the points in the first cloud are rasterized into grid cells. Each grid cell is associated with a normal distribution that locally models the distribution of the points. The transformation is then estimated by maximizing the likelihood of the second point cloud after being mapped to the space of the first cloud with regard to (w.r.t.) the normal distributions in the first cloud. The computational complexity of NDT is $\mathcal{O}(KD\log(KD))$, where K is the number of grid cells. Unfortunately, the performance of NDT is sensitive to the chosen grid size. To mitigate this issue, the Coherent Point Drift (CPD) method proposed in [3] describes the entire point cloud using a Gaussian mixture model (GMM) instead of manually allocating the points into grid cells. The resulting computational complexity of CPD is $\mathcal{O}(MD)$. As an alternative, both point clouds are modelled by GMMs in [4], [5]. The transformation is then determined by maximizing the similarity between the first GMM after transformation and the second one.

On the other hand, filtering-based methods deal with the noise and outliers by formulating the point registration problem as a filtering problem [6]–[9]. In the filters, the states are the updated transformations as the algorithms proceed, and the observation is fixed as the reference point cloud (i.e., the second point cloud). White Gaussian noise is added to both the state transition and the observation model such that the noise and outliers in the point clouds can be well described. Due to the nonlinearity in the observation model, unscented

[‡]Corresponding author: XJIANG007@e.ntu.edu.sg.
The research was partially supported by the ST Engineering – NTU Corporate Lab through the NRF corporate lab@university scheme. Hang Yu is supported by MOE (Singapore) Tier 2 project MOE2017-T2-2-126.

Kalman filters [6], particle filters [7], cubature Kalman filters (CFK) [8], and split covariance intersection filter (CSCIF) [9] are exploited to update the transformation given the reference point cloud. Although filtering-based methods achieve higher accuracy than distance-optimization-based methods, they are more computationally burdensome. For instance, the computational complexity of CKF and CSCIF is $\mathcal{O}(M^3D^3)$. The high computational cost is the biggest hurdle in applying filtering-based methods to real-world SLAM problems.

In this work, we propose a novel state-space model (SSM) for point set registration. To handle the noise and outliers, we first describe all point clouds by GMMs as in the probability-based methods. We then regard the transformations across time as the latent states, and assume that the current transformation can be decomposed as the sum of the previous transformation, the previous acceleration on both translation and rotation, and white Gaussian noise. Given the prior information, we further assume that the shape (i.e., the density) of the GMM of the current point cloud equals the shape of the GMM of the previous point cloud after transformation plus white Gaussian noise, and refer to it as the observation model. In other words, by solving the SSM, we aim to find the transformation that maximizes the similarity of the two GMMs given the prior information of the transformation. Instead of checking the similarity of the two GMMs in the entire point space, we use the quadrature points as representatives, thus further reducing the computational effort. To learn the latent states given observations, we develop an efficient stochastic variational Bayes inference algorithm. The complexity of the proposed method is $\mathcal{O}(LP)$, where L is the number of quadrature points which is a small number in 2D and 3D spaces ($L < 100$). The proposed SSM operates differently from the existing filtering-based methods [6]–[9] which artificially treat the iteration number in their algorithms as time.

We simulate data from both 2D and 3D registration scenarios, and benchmark the proposed method against the ICP [1], NDT [2], CPD [3], and CSCIF [9] method. The results demonstrate that the proposed method achieves higher accuracy with comparable or less computational time and resources in comparison with the benchmark methods.

The rest of this paper is organized as follows: the Bayesian formulation of the registration problem is introduced in Section II; the proposed SVBI solution is presented in Section III; experimental results are shown in Section IV; we offer concluding remarks in Section V.

II. Bayesian Formulation of Point Registration

Given the model set $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M\}$ and the scene set $\boldsymbol{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_N\}$ in $\mathbb{R}^D$, the objective of point registration is to find the relative transformation between the two sets, which is defined as $\boldsymbol{Z} = \mathcal{T}(\boldsymbol{U}, \boldsymbol{x})$. Specifically, $\boldsymbol{x}$ is the transition vector and defined as $[\boldsymbol{x}^t; \boldsymbol{x}^\theta]$, where $\boldsymbol{x}^t$ is the translation vector while $\boldsymbol{x}^\theta$ encodes the rotation information. As such, for every point $\boldsymbol{u}_i$ in $\boldsymbol{U}$, its correspondence in the space of Z given the alignment $\boldsymbol{x}$ can be computed as $R(\boldsymbol{x}^\theta)\boldsymbol{u}_i + \boldsymbol{x}^t$, where R is the rotation matrix that is a nonlinear function of $\boldsymbol{x}^\theta$.

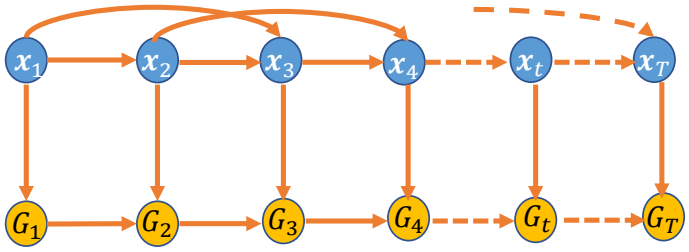


Fig. 1. Dynamic Bayesian Network of the point sets generation in a sequential system.

To overcome the disadvantages of point-wise registration, we adopt the probability-based methods and describe each finite point set using a GMM as in [4], [5]. As a result, the distribution of the point cloud $\boldsymbol{U}$ and $\boldsymbol{Z}$ can be respectively expressed as $\mathcal{G}_U(\boldsymbol{u}) = \sum_{i=1}^{m} \omega_i \mathcal{N}(\boldsymbol{u}; \boldsymbol{\mu}_i, \Sigma_i)$ and $\mathcal{G}_Z(\boldsymbol{z}) = \sum_{j=1}^{n} \lambda_j \mathcal{N}(\boldsymbol{z}; \boldsymbol{\nu}_j, \Xi_j)$, where $\mathcal{N}(\boldsymbol{u}; \boldsymbol{\mu}_i, \Sigma_i)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_i$ and covariance Σ_i and ω_i λ_i denote the weight of component i. The GMMs are obtained by clustering the points in a cloud into several Gaussian-distributed components [10]. Now, the registration between $\boldsymbol{Z}$ and $\boldsymbol{U}$ amounts to aligning $\mathcal{G}_Z$ to $\mathcal{G}_U$. As mentioned in [3]–[5], such probabilistic representations are robust to noise and outliers in the point sets.

Furthermore, in the case where the point cloud is very dense (i.e., M and N are large), characterizing the point cloud by a mixture of several Gaussian distributions can provide an efficient but reliable tool for downsampling the point set.

After modelling the point clouds by GMMs, let us turn our attention to construct an SSM for point set registration across time. Suppose that the point sets are all generated from a sequential system, e.g., the Lidar-based SLAM. The proposed SSM can then be expressed as:

$$\boldsymbol{x}_t = 2\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2} + \boldsymbol{\delta}_t \tag{1a}$$

$$\mathcal{G}_Z(\boldsymbol{y}) = \mathcal{T}(\mathcal{G}_U, \boldsymbol{x}_t)(\boldsymbol{y}) + \epsilon_t(\boldsymbol{y}), \quad \forall \boldsymbol{y} \in \mathbb{R}^D \tag{1b}$$

where $\mathcal{T}(\mathcal{G}_U, \boldsymbol{x})(\boldsymbol{y}) = \sum_{i=1}^{m} \omega_i \mathcal{N}(\boldsymbol{y}; R\mu_i + \boldsymbol{t}, R\Sigma_i R^T)$ is the density function of the GMM for $\mathcal{G}_U$ after the transformation $\mathcal{T}$, $\boldsymbol{y}$ is the argument of the density function, $\boldsymbol{\delta}_t$ and $\epsilon_t(\boldsymbol{y})$ denotes motion and observation noise. Note that the motion function (1a) can be equivalently written as

$$\boldsymbol{x}_t - \boldsymbol{x}_{t-1} = \boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2} + \boldsymbol{\delta}_t, \tag{2}$$

where $\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}$ can be regarded as the acceleration at time $t-1$. We can tell from the above expression that the motion model favors the case where the acceleration is time invariant. Viewed another way, it prefers $2\boldsymbol{x}_{t-1} - \boldsymbol{x}_t - \boldsymbol{x}_{t-2}$ to be close to zero, which is the curvature at $\boldsymbol{x}_{t-1}$. As such, $\boldsymbol{x}_t$ will change smoothly across time. Such priors are frequently used for smoothing signals in the literature of statistical signal processing [11]. On the other hand, the observation model can be interpreted as the density of the point set $\boldsymbol{Z}$ equals the density of transformed $\boldsymbol{U}$ plus independent Gaussian noises at all points $\boldsymbol{y} \in \mathbb{R}^D$. Assuming that $\boldsymbol{\delta}_t \sim \mathcal{N}(\boldsymbol{0}; \alpha_t^{-1} I_P)$ and $\epsilon_t(\boldsymbol{y}) \sim \mathcal{N}(0; \beta_t^{-1})$, where α_t and β_t are the inverse variances and I_P is a $P \times P$ identity matrix, the maximum a posterior estimate of $\boldsymbol{x}$ is given by:

$$\boldsymbol{x} = \operatorname{argmin} \alpha_t \|\boldsymbol{x}_t - 2\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t-2}\|_2^2$$

$$+ \beta_t \int \left(\mathcal{G}_Z(\boldsymbol{y}) - \mathcal{T}(\mathcal{G}_U, \boldsymbol{x}_t)(\boldsymbol{y})\right)^2 d\boldsymbol{y}. \tag{3}$$

In other words, by solving the SSM, we select the transformation $\boldsymbol{x}$ that maximizes the similarity between the two GMMs while considering the smoothness of $\boldsymbol{x}_t$ across time. We notice that the second term in (3) is the L2 distance between two distributions. It has been pointed out in [4] that L2 distance is more robust to outliers and local minima than other distances, such as KL divergence. Although the integration in the L2 distance is tractable, it is both easier and computationally feasible to approximate this integration numerically, as well as estimate the inverse variance β_t following the Bayesian paradigm. Explicitly, we utilize the Gauss-Hermite quadrature rule, which is shown to be accurate for the integration over GMMs in [12]. Suppose that we use L quadrature points $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_L\}$, whose associated weights are $\{w_1, \cdots, w_L\}$. For brevity, we use the following notation: $\boldsymbol{z}_t = [\mathcal{G}_z(\boldsymbol{y}_1), \cdots, \mathcal{G}_z(\boldsymbol{y}_L)]^T$, $h_t(\boldsymbol{x}_t) = [\mathcal{T}(\mathcal{G}_U, \boldsymbol{x}_t)(\boldsymbol{y}_1), \cdots, \mathcal{T}(\mathcal{G}_U, \boldsymbol{x}_t)(\boldsymbol{y}_L)]^T$, and $\boldsymbol{w} = [w_1, \cdots, w_L]$. As such, the observation model can be written as:

$$\boldsymbol{z}_t = h_t(\boldsymbol{x}_t) + \boldsymbol{\iota}_t, \tag{4}$$

where $\boldsymbol{\iota}_t \sim \mathcal{N}(\boldsymbol{0}; \beta_t^{-1}[\mathrm{diag}(\boldsymbol{w})]^{-1})$, and $\mathrm{diag}(\boldsymbol{w})$ is a $L \times L$ diagonal matrix with $\boldsymbol{w}$ on the diagonal. Note that the quadrature points downsample the observation space and so will speed up the inference process. Similarly, existing methods [1]–[3], [8], [9] also downsample the point sets to make the algorithm scalable to the dense cases. As mentioned in Section I, these methods allocate the points into grid cells and replace the points in a cell by the mean of the point distribution in this cell. Such downsampling approach, however, is sensitive to the grid size and the noise in the point sets. By contrast, the proposed method can handle noisy data, thus model the point sets better.

Accordingly, both the motion and observation function can be described from a probabilistic perspective:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_k) \propto \alpha_k^{\frac{P}{2}} \exp\left\{ -\frac{\alpha_t}{2} \right.$$
$$\left. (\boldsymbol{x}_t - (2\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}))^T (\boldsymbol{x}_t - (2\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2})) \right\}, \tag{5}$$

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t, \beta_t) \propto$$
$$\beta_t^{\frac{L}{2}} \exp\left\{ -\frac{\beta_t}{2} (\boldsymbol{z}_t - h(\boldsymbol{x}_t))^T \mathrm{diag}(\boldsymbol{w}) (\boldsymbol{z}_t - h(\boldsymbol{x}_t)) \right\}. \tag{6}$$

In (6), if we consider all $\boldsymbol{y} \in \mathbb{R}^D$ and compute the L2 distance analytically, $L \to \infty$ and so the observation distribution is not well defined. Since we intend to infer β_t from the data, we resort to the above quadrature rule to choose L points from $\mathbb{R}^D$.

III. Stochastic Variational Bayes Inference

Given the SSM, our ultimate goal is to estimate the transformation $\boldsymbol{x}_t$ and the noise parameters α_t and β_t given the observations $\boldsymbol{z}_t$ and the previous states $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_{t-2}$. Concretely, we exploit the empirical Bayes framework, that is, the point estimates of α_t and β_t is yielded by maximizing their likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)$ and then the posterior distribution of $\boldsymbol{x}_t$ can be expressed as $p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)$. Unfortunately, both of these two distributions cannot be computed analytically, due to the strong nonlinearity in the observation model (4). To get around this problem, we maximize the evidence lower bound (ELBO) $\mathcal{L}$ of the likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)$ instead:

$$\mathcal{L} = \int q(\boldsymbol{x}_t) \log \frac{p(\boldsymbol{z}_t, \boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)}{q(\boldsymbol{x}_t)} d\boldsymbol{x}_t \tag{7}$$
$$\leq \log p(\boldsymbol{z}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t) \tag{8}$$

The equality holds if and only if

$$q(\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t).$$

As mentioned before, $p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)$ is not available. Thus, we follow the variational Bayes approach and seek a tractable $q(\boldsymbol{x}_t)$ and point estimates of α_t and β_t to maximize $\mathcal{L}$. More specifically, we set $q(\boldsymbol{x}_t)$ to be Gaussian, that is, $q(\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_t, C_t C_t^T)$, where C_t is the Cholesky decomposition of the covariance of $q(\boldsymbol{x}_t)$. As a result, we can equivalently parameterize $\boldsymbol{x}_t$ as:

$$\boldsymbol{x}_t = C_t \boldsymbol{e} + \boldsymbol{\mu}_t, \tag{9}$$

where $\boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, I_P)$ follows a standard normal distribution. Here, we leverage the mean-field approximate and assume that $q(\boldsymbol{x}_t)$ has a diagonal covariance matrix. As a consequence, C_t is also a diagonal matrix. The resulting ELBO can then be written as:

$$\mathcal{L} = \mathbb{E}_q[\log p(\boldsymbol{z}_t, \boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t)] + \mathbb{H}[q(\boldsymbol{x}_t)] \tag{10}$$
$$= \mathbb{E}_q[\log(\det(C_t) p(\boldsymbol{z}_t, C_t \boldsymbol{e} + \mu_t)|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \alpha_t, \beta_t))]$$
$$+ \log\det C_t + \mathbb{H}[\phi(\boldsymbol{e})], \tag{11}$$

where $\mathbb{E}_q[g(\boldsymbol{x}_t)]$ denotes the expectation of $g(\boldsymbol{x}_t)$ over $q(\boldsymbol{x}_t)$, $\mathbb{H}$ denotes the entropy of a distribution, and $\phi(\boldsymbol{e}$ is the density function of $\boldsymbol{e}$. In (11), $\mathbb{H}[\phi(\boldsymbol{e})]$ is a constant, and $\log\det C_t$ can be simplified as $\log\det C_t = \sum_{i=1}^P \log[C_t]_{ii}$, since C_t is a diagonal matrix.

In order to maximize the ELBO $\mathcal{L}$, we follow the gradient ascent method. Therefore, we compute the gradient of $\mathcal{L}$ w.r.t. $\boldsymbol{\mu}_t$, C_t, α_t, and β_t as:

$$\nabla_{\boldsymbol{\mu}_t} \mathcal{L} = \mathbb{E}_{\phi(\boldsymbol{e})} \left[-\alpha_t \Theta_t + \beta_t \left(\nabla_{\boldsymbol{x}_t} h_t(\boldsymbol{x}_t)\right)^T \Psi_t \right] \tag{12a}$$
$$\nabla_{C_t} \mathcal{L} = \mathbb{E}_{\phi(\boldsymbol{e})} \left[\left(-\alpha_t \Theta_t + \beta_k (\nabla_{\boldsymbol{x}_t} h_t(\boldsymbol{x}_t))^T \Psi_t\right) \circ e \right] + \Delta_{C_t} \tag{12b}$$
$$\nabla_{\alpha_t} \mathcal{L} = \frac{P}{2\alpha_t} - \frac{1}{2} \mathbb{E}_{\phi(\boldsymbol{e})} \left[\Theta_t^{\mathrm{T}} \Theta_t \right] \tag{12c}$$
$$\nabla_{\beta_k} \mathcal{L} = \frac{L}{2\beta_t} - \frac{1}{2} \mathbb{E}_{\phi(\boldsymbol{e})} \left[\Psi_t^{\mathrm{T}} \Psi_t \right], \tag{12d}$$

where $\circ$ denotes elementwise product, Δ_{C_t} denotes a diagonal matrix with $[\frac{1}{C_{t\,11}}, \cdots, \frac{1}{C_{t\,PP}}]$ on the diagonal, $\Theta_t = \boldsymbol{x}_t - 2\boldsymbol{x}_{t-1} + \boldsymbol{x}_{t-2}$, and $\Psi_t = \sqrt{\mathrm{diag}(\boldsymbol{w})} \left(\boldsymbol{z}_t - h_t(\boldsymbol{x}_t)\right)$. The above gradients, however, cannot be calculated in closed form, due to the intractable expectations. Instead, we replace the exact gradients in (12a) by stochastic gradients to maximize $\mathcal{L}$. Stochastic gradients are unbiased estimates of the exact

Algorithm 1: Stochastic variational Bayes inference.

Data: $\boldsymbol{z}_t, \boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}$
Result: $\boldsymbol{x}_t$
Initialize $\boldsymbol{\mu}_t^(0), C_t^0, \alpha_t^0, \beta_t^0$, k=0 ;
while *convergence criterion is not met* **do**
  k=k+1;
  $\boldsymbol{e} \sim \phi(\boldsymbol{e})$;
  $\boldsymbol{x}_t^k = C_t^k \boldsymbol{e} + \boldsymbol{u}_t^k$;
  For a variable $\nu_t \sim \{\boldsymbol{\mu}_t, U_t, \alpha_t, \beta_t\}$,
  Calculate its stochastic gradient $(\nabla_\nu \mathcal{L})^{(k)}$;
  $\nu_t^{(k+1)} = \nu_t^{(k)} + \rho_t^{(k)} (\nabla_\nu \mathcal{L})^{(k)}$;
end

gradient that can be evaluated in a computationally cheap manner. More precisely, we provide unbiased estimates of the expectations over $\phi(\boldsymbol{e})$ by drawing one sample from $\phi(\boldsymbol{e})$ and evaluating the value of the functions inside the expectation based on this sample. As an example, the unbiased stochastic gradient corresponding to (12a) can be written as:

$$\nabla_{\boldsymbol{\mu}_t} \mathcal{L} \approx -\alpha_t \Theta_t + \beta_k \nabla_{\boldsymbol{x}_t} h_t \boldsymbol{x}_t{}^T \Psi_t \tag{13}$$

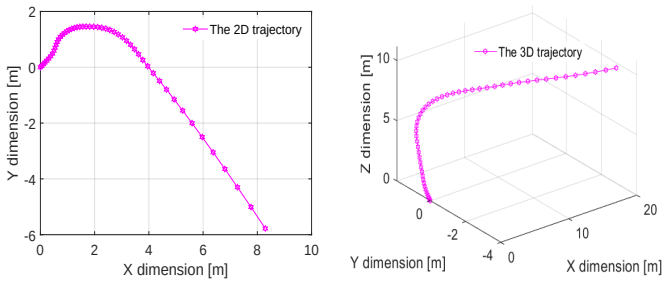
The stochastic gradients of C_t, α_t, β_t can be computed in a similar fashion. Equipped with the stochastic gradients, we maximize $\mathcal{L}$ following Algorithm 1, where $\rho_t^{(k)}$ is the step size in iteration k. According to the Robbins-Monro theorem [13], the stochastic gradient ascent algorithm in Algorithm 1 is guaranteed to converge if the step sizes satisfy $\sum_k \rho_t^{(k)} = \infty$ and $\sum_k {\rho_t^{(k)}}^2 < \infty$. In practice, we employ ADAM [14] to determine the step in every iteration. Also, we adopt the control variate in [15] and remove the score function term in the gradients as in [16] so as to reduce the variance of the stochastic gradients and to further speed up the convergence. The computational complexity of the proposed method is $\mathcal{O}(LP)$. One merit of using a stochastic gradient method is that it can easily escape from shallow local maxima [18], as the stochastic variational Bayes inference algorithm can be interpreted as a simulated annealing-type method. As opposed to the proposed method, the existing methods [1], [4], [6]–[9] are susceptible to local maxima. Furthermore, in comparison with the filtering-based methods [6]–[9], the computational complexity of the proposed method is much lower, thus making it suitable for real-time applications. After obtaining the transformation $\boldsymbol{x}_t$ at time t, we then proceed to time $t+1$. The overall variational Bayesian point set registration (VB-PSR) algorithm is summarized in Algorithm 2.

IV. Experimental Results

In this section, we benchmark the proposed VB-PSR method with the commonly-used ICP [1] and NDT [2], the representative probability-based method CPD [3], and the recent filtering-based method CSCIF [9]. We simulate synthetic data from the proposed SSM in (1). Specifically, the initial translation and angular velocity on each axis at $t = 1$ are set to be $0.05\ m$ and $0.02\ rad$, while the translation and angular acceleration on each axis are set to be $0.005\ m/(\Delta t)^2$ and $0.002\ rad/(\Delta t)^2$. For the hyperparameters of the additive Gaussian noise in the motion function,

Algorithm 2: Variational Bayesian Point Set Registration (VB-PSR).

Data: Template point sets $\boldsymbol{U}_{1:T}$, Scene Point sets $\boldsymbol{Z}_{1:T}$
Result: $\boldsymbol{x}_{1:T}$
Estimate $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$;
for $t \leftarrow 3$ **to** T **do**
  Represent U_t and Z_t by GMMs $\mathcal{G}_{U_t}$ and $\mathcal{G}_{Z_t}$;
  Calculate the quadrature points of $\mathcal{G}_{Z_t}$;
  Initialize $\boldsymbol{x}_t$ as $2\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}$;
  Employ Algorithm 1 to refine the initial estimation;
  Output $\boldsymbol{x}_t$;
end

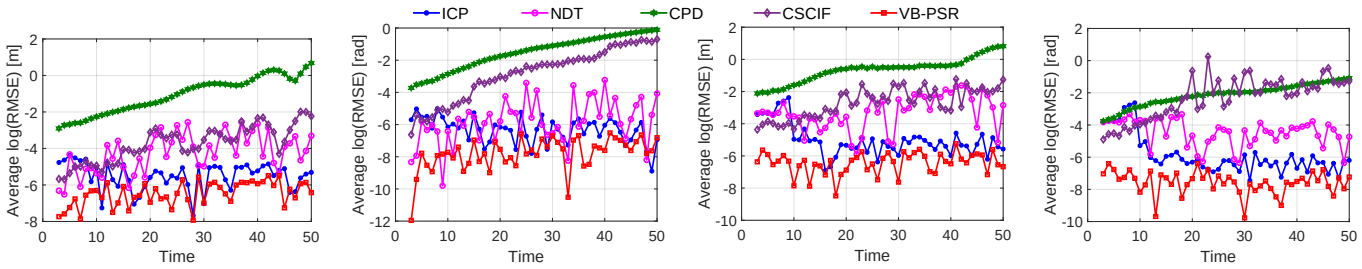


(a) (b)

Fig. 2. The ground truth of the trajectory of the vehicle in the 2D (a) and the 3D (b) case for one data set.

we set $\alpha_t = 1/(0.005\ m)^2$ for the translation vector and $\alpha_t = 1/(0.002\ rad)^2$ for the angles. We consider both 2D and 3D case. The vehicle starts from the origin point. Examples of the trajectories in these two cases are shown in Fig. 2(a) and 2(b) respectively. Next, we proceed to generate point sets. The initial GMM at $t = 1$ is assumed to have two equally-weighted components. The mean and covariance of the GMMs at time t is changed according to the rigid transformation specified by $\boldsymbol{x}_t$. We then generate the point cloud at time t by sampling from the corresponding GGM. Note that point sets produced in such manner contain noise and outliers. We set the number of points as 5×10^3 and 1×10^5 for the 2D and 3D case respectively. The number of frames $T = 50$. In particular, for NDT, we set the grid step to be $0.5\ m$. Additionally, for CSCIF, we downsample the point sets using a grid filter with grid size $1.5\ m$, since CSCIF is only applicable to low-dimensional problems with small point sets. We do the same for CPD in the 3D case using a grid filter with size $0.5\ m$. For all methods, the transformation $\boldsymbol{x}_t$ at t is initialized as $\boldsymbol{x}_{t-1} - 2\boldsymbol{x}_{t-2}$. All results presented here are averaged over 100 data sets.

We first depict the RMSE (root mean square error) between the true and estimated transformation $\boldsymbol{x}_t$ as a function of time t in Figs 3(b)-3(e). It clearly shows that the RMSE of the proposed VB-PSR achieves the smallest RMSE in almost all cases. By contrast, the RMSEs of CSCIF and CPD are several orders of magnitude and increase sharply with time t. This is probably because the downsampling, especially using a quite large grid step size, can cause a serious information loss, and further deteriorate the performance. However, the sampled quadrature points in VB-PSR can approximate the GMMs, which summarize the whole datasets, quite perfectly. On the

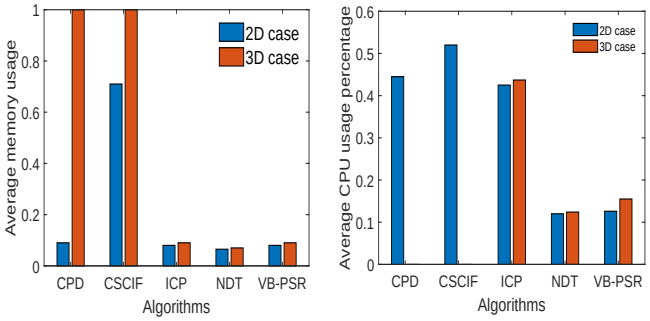


(b) Translation vector in 2D case. (c) Rotation vector in 2D case. (d) Translation vector in 3D case. (e) Rotation vector in 3D case.

Fig. 3. The logarithm of the RMSE for both the translation vector $\boldsymbol{x}^t$ and the rotation vector $\boldsymbol{x}^\theta$ as a function of time t resulting from all methods in both 2D and 3D case.

TABLE I
THE AVERAGE RUNTIME FOR ALL METHODS IN 2D AND 3D CASES.

Algorithm	ICP	NDT	CPD	CSCIF	VB-PSR
2D case	0.03 s	0.58 s	0.53 s	17.82 s	0.29 s
3D case	0.63 s	23.51 s	1.64 s	48.35 s	0.88 s



(a) Memory. (b) CPU.

Fig. 4. The average memory (a) and CPU (b) usages of all the test methods in 2D and 3D cases without downsampling. CPD and CSCIF run out of memory in the 3D case.

other hand, ICP and NDT perform better than CSCIF and CPD, since they operate on the full pointset. Unfortunately, both ICP and NDT are sensitive to the initial misalignment, and sometimes get stuck in inadequate solutions. For example, ICP has a poor performance in the beginning frames of the 3D case. In contrast, the stochastic gradient method employed in VB-PSR is more capable of escaping from the local maxima and thus having higher accuracy.

Next, we present the average registration time per frame for all methods in Table I. ICP is the fastest, due to its simplicity and low computational complexity. The proposed VB-PSR performs the second best and its computational time is almost the same with ICP in the 3D case. Note that in practice we typically deal with 3D problems instead of 2D. In contrast, CSCIF takes much longer time than other methods, due to its high computational complexity. Here, to assure the accuracy, the grid size we choose for NDT is relatively large, resulting in a heavy computational cost in the 3D case [8].

Finally, we show the memory and CPU usage of all methods without downsampling in Fig. 4. It can be observed that VB-PSR enjoys low CPU and memory usage. Thus it provides a useful tool in practice.

V. CONCLUSIONS

In this paper, we proposed a novel Bayesian state space model for point registration problem. The point sets were described by GMMs. The transformation (i.e., the states) was assumed to change smoothly across time in the SSM, while the observation was the density of the GMM at selected quadrature points given the transformation. Efficient stochastic variational Bayes inference algorithm was derived to learn the model. Numerical results showed that the proposed method can achieve higher accuracy with a reasonable amount of computational time and resources. In future work, we will test the proposed method on real-life data in intelligent vehicle localization and mapping tasks.

REFERENCES

[1] J. P. Besl, and Neil D. McKay, "Method for registration of 3-D shapes", *Sensor Fusion IV: Control Paradigms and Data Structures, International Society for Optics and Photonics*, Vol. 1611, 1992.
[2] P. Biber, and W. Strasser, "The normal distributions transform: A new approach to laser scan matching", *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, Vol. 3. IEEE, 2003.
[3] A. Myronenko, and X. Song, "Point set registration: Coherent point drift", *IEEE Trans. Pattern Anal. Mach. Intell.*, 32.12 (2010): 2262-2275.
[4] B. Jian, and B. C. Vemuri, "Robust point set registration using gaussian mixture models", *IEEE Trans. Pattern Anal. Mach. Intell.*, 33.8 (2011): 1633-1645.
[5] L. Li, et al., "Robust Point Set Registration Using Signature Quadratic Form Distance", *IEEE Trans. Cybern.*, 99 (2018): 1-13.
[6] M. H. Moghari, and Purang Abolmaesumi, "Point-based rigid-body registration using an unscented kalman filter", *IEEE Trans. Med. Imag.*, 26.12 (2007): 1708-1728.
[7] R. Sandhu, et al., "Point set registration via particle filtering and stochastic dynamics", *IEEE Trans. Pattern Anal. Mach. Intell.*, 32.8 (2010): 1459-1473.
[8] L. Li, et al., "Rigid point set registration based on cubature Kalman filter and its application in intelligent vehicles", IEEE Trans. Intell. Transp. Syst., 19.6 (2018): 1754-1765.
[9] L. Li , et al., "Cubature Split Covariance Intersection Filter-Based Point Set Registration", IEEE Trans. Image Process, 27.8 (2018): 3942-3953.
[10] G. McLachlan, and D. Peel, "Finite mixture models John Wiley Sons", *Inc. , Hoboken*, New Jersey (2000).
[11] H. Yu, and J. Dauwels, "Modeling spatio-temporal extreme events using graphical models", *IEEE Trans. Signal Process*, 64.5 (2016): 1101-1116.
[12] R. Linares, and J. Crassidis, "Sigma point transformation for gaussian mixture distributions", *AIAA Guid. Navig. Control Conf.*, 2012.
[13] H. Robbins, and S. Monro, "A stochastic approximation method", *Annals Math. Stats.*, 1985, 102-109.
[14] D. P. Kingma, and J. L. Ba, Adam: A method for stochastic optimization. *Proc. ICLR*, 2015.
[15] A. Miller, et al., "Reducing reparameterization gradient variance", Proc. NIPS., 2017.
[16] G. Roeder, et al., "Sticking the landing: Simple, lower-variance gradient estimators for variational inference", Proc. NIPS., 2017.
[17] M. Titsias, and M. Lzaro-Gredilla, "Doubly stochastic variational Bayes for non-conjugate inference", *Proc. ICML.*, 2014.
[18] X. Jiang, et al, "Linear-complexity stochastic variational Bayes inference for SLAM", *Proc. ITSC.*, IEEE, 2017.
[19] F. Pomerleau, et al., "Comparing ICP variants on real-world data sets", *Autonomous Robots*, 34.3 (2013): 133-148.