Stein ICP for Uncertainty Estimation in Point Cloud Matching

Fahira Afzal Maken , Fabio Ramos , and Lionel Ott

Abstract—Quantification of uncertainty in point cloud matching is critical in many tasks such as pose estimation, sensor fusion, and grasping. Iterative closest point (ICP) is a commonly used pose estimation algorithm which provides a point estimate of the transformation between two point clouds. There are many sources of uncertainty in this process that may arise due to sensor noise, ambiguous environment, initial condition, and occlusion. However, for safety critical problems such as autonomous driving, a point estimate of the pose transformation is not sufficient as it does not provide information about the multiple solutions. Current probabilistic ICP methods usually do not capture all sources of uncertainty and may provide unreliable transformation estimates which can have a detrimental effect in state estimation or decision making tasks that use this information. In this work we propose a new algorithm to align two point clouds that can precisely estimate the uncertainty of ICP's transformation parameters. We develop a Stein variational inference framework with gradient based optimization of ICP's cost function. The method provides a non-parametric estimate of the transformation, can model complex multi-modal distributions, and can be effectively parallelized on a GPU. Experiments using 3D kinect data as well as sparse indoor/outdoor LiDAR data show that our method is capable of efficiently producing accurate pose uncertainty estimates.

Index Terms—Perception for grasping and manipulation, localization, range sensing, probabilistic inference.
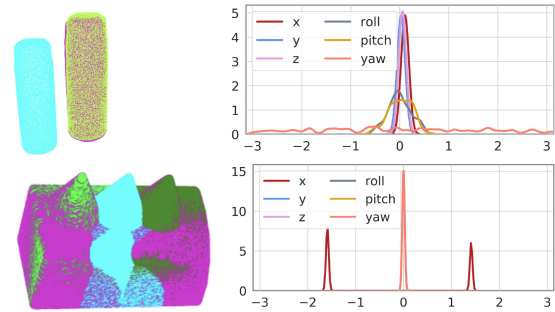
Fig. 1. Our method estimates ICP pose distributions (right) for the solution (green) when a source cloud (cyan) is aligned to a reference cloud (magenta). Flat yaw distribution (top right) characterizes the rotational ambiguity of the shape. For the bottom figure, our method estimates two possible modes for x as shown in the corresponding kernel density estimate on the right which a deterministic ICP algorithm cannot capture.

I. INTRODUCTION

POINT cloud registration plays a fundamental role in many robotics and computer vision tasks such as localization and mapping [19], autonomous navigation [50], pose estimation [41], surgical guidance [33], and augmented reality [38] to name a few. Iterative closest point (ICP) [8] is the gold standard registration algorithm which estimates a relative transformation between two point clouds. Given an initial estimate, ICP minimizes the Euclidean distance between pairs of matching points from both point clouds in an iterative manner.

The ICP pose alignment process is adversely affected by different sources of error and uncertainty. These include initial pose uncertainty, sensor noise, partial overlap, multiple local minima of the cost function, and under-constrained or ill-posed cases, *e.g.* long featureless corridors or rotational symmetric objects such as bottles that admit infinite solutions. In the alignment of an under-constrained object where there is ambiguity in the matching, uncertainty relates to the geometry of the object. This type of uncertainty is intrinsic to the problem and known as *epistemic*. For example, in Fig. 1 (top), the broad yaw distribution captures the rotational symmetry of the bottle. For a rectangular building block (bottom left), source cloud (cyan) can match to any of the two sides of the reference cloud (magenta), thus giving bi-modal x distribution (right) which a deterministic ICP algorithm cannot capture.

The quantification of the transformation uncertainty is crucial for many other tasks, especially those requiring robust pose estimates of an autonomous vehicle in an urban environment. In order to accurately localize a vehicle, a single source of pose information is usually not enough. Pose information from different sensors are usually probabilistically fused together [12], [52] in a state estimation framework such as simultaneous localization and mapping (SLAM) [4]. In such a framework, ICP pose estimates are fused with odometry measurements or GPS observations using Extended Kalman Filter (EKF) [5], [16], particle filter [47] or GraphSLAM [10] which require information about the uncertainty of pose parameters. This uncertainty estimate encompasses information of the reliability of sensory data and aids in improving the estimates of the pose parameters. In the localization and mapping tasks, this uncertainty information can improve loop closure detection and yield better maps. In the task of scene reconstruction, the amount of uncertainty in the transformation parameters can indicate a tangential drift that may occur in the registration of two plane objects.

Bayesian methods [39] can be used to incorporate uncertainty into ICP pose estimates. These methods provide a probabilistic framework in which Bayes' rule is used to obtain a posterior distribution given a prior distribution and a likelihood function. However, Bayesian methods tend to be computationally intractable for complicated likelihood functions and high-dimensional problems. To address the tractability issue, modern Bayesian methods rely on approximate techniques such as Markov Chain Monte Carlo (MCMC) [40] and variational inference (VI) [56] which can scale to large problems using stochastic gradient descent (SGD) [46].

MCMC methods approximate the intractable posterior distribution by drawing samples from the prior and likelihood functions. In contrast, VI transforms the problem into an optimization process that reduces the Kullback-Leibler (KL) divergence between the intractable posterior distribution and an analytically tractable variational distribution. MCMC will converge to the underlying distribution but can be slow in practice. VI on the other hand is typically faster but the expressiveness of the variational distribution is restricted and may not capture the complexity of the true posterior distribution.

In this letter we formulate a Stein variational gradient descent (SVGD) [30] method to approximate the posterior distribution of ICP pose parameters. SVGD combines the accuracy and flexibility of MCMC with the speed of VI using gradient descent. SVGD approximates the intractable posterior distribution with a non-parametric representation given by a set of particles. These particles are optimized with a functional gradient descent of KL divergence and provide a reliable uncertainty estimation by incorporating a *repulsive* term that prevents particles from clustering together and allows them to capture a wide range of complex distributions. These particles can be updated jointly and benefit from GPU parallelization. **Contribution:** The main contribution of this letter is a non-parametric point cloud registration method based on Stein variational gradient descent to estimate ICP's pose uncertainty. Our method exploits GPU-computation for increased speed and captures both epistemic and aleatory uncertainty [27] intrinsic in the matching problems. A python implementation of the method is publicly available.[1]

II. RELATED WORK

There exists a large number of point cloud registration methods as reviewed in [42], [48]. They can be broadly categorized as algorithms that provide point estimates and probabilistic outputs.

A. Point-Based ICP Algorithms

Point estimate registration algorithms produce a single estimate of the transformation between two point clouds. These algorithms mostly vary in how they select points to perform data association, which error metric they use, and what optimization technique they employ for their cost function [48].

ICP algorithms which vary in the selection of points include those using random sampling [36] and uniform subsampling [35]. Other ICP methods have different cost functions. Notable choices are point-to-point [8] point-to-plane [14] and plane-to-plane [49]. The ICP cost function can be optimized using either closed form solutions *e.g.* singular value decomposition (SVD) [3], and quaternions [22] or other methods

e.g. Levenberg-Marquardt [20], simulated annealing [31], and stochastic gradient descent [1].

Another line of research deals with eliminating outliers due to partial overlap, sensor noise and other complexities. This includes the use of a trimmed square cost function to estimate the optimal transformation [15], and rejecting pairs with distance larger than a certain threshold [36], [49].

An alternative class of registration technique reformulates ICP as an alignment of two Gaussian mixtures where a statistical discrepancy measure is minimized between the two mixtures [18]. In this case Expectation Maximization [17] can be used to provide robust solutions *i.e.* [21]. Learning-based point cloud registration methods employ neural networks to provide pose parameters [2], [54]. These methods rely on the data used during training and may not necessarily generalize well to an unseen dataset.

B. Uncertainty-Based ICP Algorithms

This class of ICP provides uncertainty estimates for the transformation parameters. The uncertainty of the transformation can be estimated using closed form solutions [7], [9], [11], [13], and sampling-based approaches [7], [24], [34]. Closed form solutions require the Hessian of the cost function. Among closed form solutions such as in [7], [9], [11], errors due to sensor noise are not properly captured as pointed out in [13], [44]. [13] considers the effect of sensor noise in the covariance structure but [32], [37] indicate that it is overoptimistic and hence not suitable for sensor fusion. Closed-form solutions are efficient but do not capture the uncertainty of the data association into account in the covariance estimation.

Sampling based approaches such as [7], [24] provide high quality samples of pose parameters but are computationally expensive. A recently proposed scalable probabilistic ICP method, Bayesian ICP [34], employs stochastic gradient Langevin dynamics (SGLD) [55] to provide samples of the posterior distribution in an online manner. However, due to it still being an MCMC style method it suffers from the inherent limitations and challenges of parallelizing any MCMC method.

Other recently proposed methods such as [29], [32] take initial pose estimates into account in the pose uncertainty prediction. [32] combines a closed-form solution to incorporate sensor noise with the unscented transform [25] to capture other sources of uncertainty. This method relies on an accurate covariance of the initial pose uncertainty to produce reliable uncertainty estimation which is not always available.

Our method can take prior information into account to estimate the posterior distribution over transformation parameters. When a prior is available, our method leverages this information to achieve faster and more accurate convergence. In comparison to Bayesian ICP which uses MCMC sampling, our method employs Stein particles that can be propagated in parallel through Stein's gradients, directly modeling the interactions between the particles and promoting diversity in the solution. This is particularly important in multi-modal problems as we demonstrate in a few examples. Additionally, our method is directly amenable to parallelization, therefore suitable to GPU computation.

III. PRELIMINARIES

In this section we first give an overview of standard ICP [8], followed by a description of stochastic gradient descent ICP

---

[1][Online]. Available: https://bitbucket.org/fafz/stein-icp/src/master/

(SGD-ICP) [1] which constitutes a core component of our method.

A. Standard ICP

Standard ICP aligns two input point clouds by iteratively performing the following two steps until convergence:

1) Establish the point pairings on the basis of minimum Euclidean distance using an initial guess $\boldsymbol{\theta} = \{x, y, z, \text{roll}, \text{pitch}, \text{yaw}\}$. The commonly used point-to-point distance metric to select the corresponding pair of points is expressed as follows:

$$\text{point-to-point}(\mathbf{s_i}', \mathscr{R}) = \min_{\mathbf{r} \in \mathscr{R}} \|\mathbf{r} - \mathbf{s_i}'\|, \quad (1)$$

where $\mathbf{s_i}, \mathbf{r_i} \in \mathbb{R}^3$ are points in 3D space belonging to a source cloud $\mathscr{S} = \{\mathbf{s_i}\}_{i=1}^N$ and a reference cloud $\mathscr{R} = \{\mathbf{r_i}\}_{i=1}^M$ respectively. $\mathbf{s_i}' = (R\,\mathbf{s_i} + \mathbf{u})$ is a transformed point in the source cloud, $\mathbf{u} \in \mathbb{R}^{3 \times 1}$ is a translation vector comprising of $\boldsymbol{\theta}_{1:3}$ and $R \in \mathbb{R}^{3 \times 3}$ is a rotation matrix parametrized by $\boldsymbol{\theta}_{4:6}$.

2) Find the transformation that minimizes a loss function defined by the Euclidean distance between each pair of corresponding points. The point-to-point cost function has the following form:

$$\arg\min_{\boldsymbol{\theta}} \mathscr{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_i^N \|(R\,\mathbf{s_i} + \mathbf{u}) - \mathbf{r_i}\|^2. \quad (2)$$

B. Stochastic Gradient Descent ICP

SGD-ICP [1] optimizes the ICP cost function (2) using stochastic gradient descent (SGD) [46]. Instead of using all available points, SGD-ICP samples a mini-batch $\mathscr{M}$ of m points from the source cloud to find corresponding points from the reference cloud. SGD-ICP computes gradients of the cost function (2) to update the transformation parameters θ with the following update rule:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta A \bar{g}(\boldsymbol{\theta}^t, \mathscr{M}^t), \quad (3)$$

where $\boldsymbol{\theta}^t$ and $\boldsymbol{\theta}^{t+1}$ are the values of the pose parameters at the current and next iteration respectively. The matrix $A \in \mathbb{R}^{6 \times 6}$ acts as a pre-conditioner for the gradients and the learning rate η dictates how quickly parameter values change. The average gradients $\bar{g}$ of the parameter vector $\boldsymbol{\theta}$ with respect to the loss $\mathscr{L}$ are computed as follows:

$$\bar{g}(\boldsymbol{\theta}_{1:3}^t, \mathscr{M}^t) = \frac{1}{m} \sum_i^m \left((R^t\,\mathbf{s_i} + \mathbf{u}^t) - \mathbf{r_i}\right) \frac{\partial \mathbf{u}^t}{\partial \boldsymbol{\theta}_{1:3}^t}, \quad (4)$$

$$\bar{g}(\boldsymbol{\theta}_{4:6}^t, \mathscr{M}^t) = \frac{1}{m} \sum_i^m \left((R^t\,\mathbf{s_i} + \mathbf{u}^t) - \mathbf{r_i}\right) \frac{\partial R^t}{\partial \boldsymbol{\theta}_{4:6}^t}\,\mathbf{s_i}, \quad (5)$$

where $\bar{g}(\boldsymbol{\theta}_{1:3}^t, \mathscr{M}^t)$ are the gradients of the translation components and $\bar{g}(\boldsymbol{\theta}_{4:6}^t, \mathscr{M}^t)$ are the rotational gradients.

The quality of the estimated pose of SGD-ICP is the same as that of standard ICP but it is computationally more efficient due to the mini-batch formulation [1]. In the next section we employ the mini-batch gradient-based formulation of SGD-ICP within Stein variational inference to derive Stein ICP.

IV. STEIN ICP

Our proposed method, Stein ICP, makes the connection between the gradient-based stochastic optimization of an ICP cost function and Stein variational inference. Specifically, Stein ICP utilizes the SGD-ICP [1] gradients within the Stein variational gradient descent (SVGD) framework [30]. SVGD approximates an intractable but differentiable posterior distribution $p(\mathbf{x})$ by constructing a non-parametric variational distribution represented by a set of K particles $\{\mathbf{x}_j\}_{j=0}^K$. These particles are updated iteratively by an update rule of the following form:

$$\mathbf{x}_j \leftarrow \mathbf{x}_j + \eta \phi(\mathbf{x}_j) \quad \forall j = 1, \dots, K, \quad (6)$$

where η is a small step size, and $\phi : \mathbb{R}^d \to \mathbb{R}^d$ characterizes a perturbation direction that gives the steepest descent. ϕ should be chosen to push-forward the particle distribution closer to the target. The optimal value of the perturbation direction ϕ can be obtained from the following expression:

$$\phi^* = \underset{\phi \in \mathscr{B}}{\arg\max} \left(-\frac{d}{d\eta} \text{KL}(q_{[\eta\phi]} \| p)\big|_{\eta=0} \right), \quad (7)$$

where $q_{[\eta\phi]}$ is specified nonparametrically by the updated particles, $\mathbf{x}' = \mathbf{x} + \eta\phi(\mathbf{x})$. ϕ^* is selected from a function set, $\mathscr{B}$, to minimize the KL divergence between the particle distribution and the target [53].

To obtain tractable and flexible solutions, SVGD chooses $\mathscr{B}$ to be in the unit ball of a vector-valued reproducing kernel Hilbert space (RKHS) $\mathscr{H}^d = \mathscr{H} \times \cdots \times \mathscr{H}$, where $\mathscr{H}^d$ is an RKHS formed by scalar-valued functions associated with a positive definite kernel $k(\mathbf{x}, \mathbf{x}')$, that is, $\mathscr{B} = \{\phi \in \mathscr{H}^d : \|\phi\|_{\mathscr{H}^d} \leq 1\}$. Formally, an RKHS is defined as a Hilbert space $\mathscr{H}^d$ with inner product $\langle \cdot, \cdot \rangle_{\mathscr{H}^d}$ and norm $\|.\|_{\mathscr{H}^d}$ satisfying the reproducing property, $\forall f \in \mathscr{H}^d, f(\mathbf{x}) = \langle f, k(\mathbf{x}, .) \rangle_{\mathscr{H}^d}$. A key observation to solve Eq. 7 is that the optimization is a linear functional of ϕ that connects to the Stein operator $\mathscr{A}$ in the following equation [30],

$$-\frac{d}{d\eta} \text{KL}(q_{[\eta\phi]} \| p)\big|_{\eta=0} = \mathbb{E}_{\mathbf{x} \sim q}[trace(\mathscr{A}\phi(\mathbf{x}))], \quad (8)$$

with

$$\mathscr{A}\phi(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})\phi(\mathbf{x})^\top + \nabla_{\mathbf{x}}\phi(\mathbf{x}). \quad (9)$$

By Stein's identity in Eq. 8, we obtain $\frac{d}{d\eta}\text{KL}(q_{[\eta\phi]} \| p) = 0$ when $p = q$. Following from (9), the gradient of the log posterior $\nabla_{\mathbf{x}} \log p(\mathbf{x}|D)$ can be expressed as:

$$\nabla_{\mathbf{x}|D} \log p(\mathbf{x}) = \sum_{i=1}^N \nabla_{\mathbf{x}} \log p(d_i|\mathbf{x}) + \nabla_{\mathbf{x}} \log p(\mathbf{x}), \quad (10)$$

where $\nabla_{\mathbf{x}} \log p(d_i|\mathbf{x})$ and $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ are the gradient of the log likelihood function, and the gradient of the log of the prior with respect to $\mathbf{x}$ respectively, for a dataset $D = \{d_i\}_{i=1}^N$ of N data points. Note that the Stein operator does not depend on the normalization $p(D)$, also known as the marginal likelihood, which makes the computation of the posterior significantly easier. This makes SVGD a powerful tool for inference of intractable distributions. The Stein variational gradient descent algorithm follows from a closed form solution of (7), as shown in [30], and given by:

$$\phi^*(\cdot) = \mathbb{E}_{\mathbf{x} \sim q}[\nabla_{\mathbf{x}} \log p(\mathbf{x})k(\mathbf{x}, \cdot) + \nabla_{\mathbf{x}}k(\mathbf{x}, \cdot)]. \quad (11)$$

In the above we omit the dependence on the data D to simplify notation. Equation (11) provides the optimal update direction for the particles within $\mathcal{H}^d$. In practice, a set of particles $\{\mathbf{x}_j\}_{j=1}^{K}$ approximates q yielding the following update rule:

$$\mathbf{x}_j \leftarrow \mathbf{x}_j + \eta \hat{\boldsymbol{\phi}}^*(\mathbf{x}_j), \qquad (12)$$

where $\hat{\boldsymbol{\phi}}^*(\mathbf{x})$ is an approximate steepest direction given by

$$\hat{\boldsymbol{\phi}}^*(\mathbf{x}) = \sum_{j=1}^{K}[\nabla \log p(\mathbf{x}_j)k(\mathbf{x}_j, \mathbf{x}) + \nabla_{\mathbf{x}_j}k(\mathbf{x}_j, \mathbf{x})]. \qquad (13)$$

The first gradient term in (13) is weighted by a kernel function, which smooths the gradients, and acts as the steepest direction for the log probability. The second term is the gradient of the kernel function. It can be viewed as a repulsive force which induces the spread among the particles and prevents them from collapsing to the local modes of the log probability.

In SGD-ICP $-\nabla_{\mathbf{x}} \log p(D|\mathbf{x})$, in Eq. 10, is expressed as the gradients of the cost function in (4) and (5). $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ represents the gradient of the log of Gaussian priors for translations and von Mises priors for rotations [34]:

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_{1:3}) = -(\boldsymbol{\theta}_{1:3} - \boldsymbol{\mu}_{1:3})/\boldsymbol{\sigma}_{1:3}, \qquad (14)$$

$$\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_{4:6}) = -\boldsymbol{\kappa}_{4:6}\sin(\boldsymbol{\theta}_{4:6} - \boldsymbol{\mu}_{4:6}), \qquad (15)$$

where $\boldsymbol{\mu}_{1:6}$ represents the mean while $\boldsymbol{\sigma}_{1:3}$, and $\boldsymbol{\sigma}_{4:6} = 1/\boldsymbol{\kappa}_{4:6}$ represent the variance of the prior distributions of the translation and rotation components respectively.

In Stein ICP, we replace the gradients in (13) with the mini-batch gradients of the ICP cost function shown in (4) and (5), along with the gradients of priors expressed in (14) and (15). By doing this, Stein variational gradients are independently obtained for translations ($\boldsymbol{\theta}_{1:3}$) and rotations ($\boldsymbol{\theta}_{4:6}$), for all the particles in (16), which are then updated using (12).

$$\hat{\boldsymbol{\phi}}^*(\boldsymbol{\theta}) = \sum_{j=1}^{K} \Big[-\big(N\bar{g}(\boldsymbol{\theta}_j, \mathcal{M}) + \nabla_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta}_j)\big)k(\boldsymbol{\theta}_j, \boldsymbol{\theta})$$
$$+ \nabla_{\boldsymbol{\theta}}k(\boldsymbol{\theta}_j, \boldsymbol{\theta})\Big]. \qquad (16)$$

In SVGD the kernel plays a critical role in weighing the gradients and dispersing the particles. In Stein ICP, for translation parameters we use an RBF kernel $k(\boldsymbol{\theta}_{1:3}, \boldsymbol{\theta}'_{1:3}) = \exp(-\frac{1}{h}||\boldsymbol{\theta}_{1:3} - \boldsymbol{\theta}'_{1:3}||_2^2)$, where h is the bandwidth of the kernel. For the rotational parameters, we use a modified version of the RBF kernel as the Euclidean distance is not directly applicable to angles. In particular we use the following modified RBF kernel for rotations:

$$k(\boldsymbol{\theta}_{4:6}, \boldsymbol{\theta}'_{4:6}) = \exp\Bigg[-\frac{1}{h}\bigg(\arctan 2(\sin(\boldsymbol{\theta}_{4:6} - \boldsymbol{\theta}'_{4:6}),$$
$$\cos(\boldsymbol{\theta}_{4:6} - \boldsymbol{\theta}'_{4:6}))\bigg)^2\Bigg]. \qquad (17)$$

In summary, Stein ICP begins by initializing a set of K particles randomly. In general the implementation of Stein variational inference does not depend on the initial distribution [30]. In practice, the particles can be initialized using prior knowledge, *e.g.* readings from an inertial measurement unit or from a previous solution. Each particle represents the transformation of a mini-batch sampled from a source cloud $\mathscr{S}$ to a reference cloud $\mathscr{R}$, producing K transformed mini-batches. Next for all points in each transformed mini-batch, corresponding closest points from a reference cloud are sought and stored in pairs using (1). Then mean gradients are estimated for all the matching pairs belonging to each of the K particles using (4) and (5). Next, Stein variational gradients are obtained independently for translations and rotations using (16) which are then used to update each particle with (12). This procedure is repeated for T iterations producing K particles representing the posterior distribution.

V. EXPERIMENTS

In the experiments we demonstrate the ability of Stein ICP to obtain high quality pose distributions while its formulation leads to significant computational gains when using a GPU.

To gain an intuitive insight into the behaviour of Stein ICP, we use objects with obvious distributions from an RGB-D dataset [28]. For the quantitative evaluations we use a challenging point cloud alignment dataset [43] which contains different sequences recorded in both structured and unstructured environments. We provide comparisons with three other uncertainty aware ICP methods, namely Bayesian ICP [34], Cov-3D ICP [32], and Closed-form ICP [13].

As there is no ground truth distribution available for the point cloud alignment we compute this ground truth information using Monte Carlo sampling. To obtain these Monte Carlo samples in a timely manner, we use SGD-ICP, which produces solutions equivalent to standard ICP methods in quality but much more quickly [1]. We run 1000 instances of SGD-ICP with initial transformation estimates sampled from $\pm 1\,\mathrm{m}$ and $\pm 0.1745\,\mathrm{rad}$ for translations and rotations respectively. The resulting samples form the ground truth distribution. As a measure of the quality of the distribution estimated by a method we compute the Kullback-Leibler (KL) divergence and overlapping coefficient (OVL) [23] between the obtained distribution and the ground truth distribution. A value of 0 and 1 for OVL indicates no-overlap and full overlap between the two probability distributions respectively.

In all experiments, Stein ICP uses Adam [26] as the optimizer with a step size of 0.03 and batch size of 150 for the RGB-D data and a step size of 0.01 and batch size of 300 for the Challenging dataset [43]. The bandwidth parameter required by SVGD is chosen using the median heuristic as described in [30]. Furthermore, uniform priors are assumed as no prior knowledge is available. Bayesian ICP generates 1000 samples and uses a step size of 0.008. Cov-3D and Closed-form ICP use point-to-plane SGD-ICP estimated pose to compute the covariance matrix using the default parameter setting of the author's implementation [32].

A. Particle Count

We begin by evaluating the effect the number of particles has on the pose distribution quality on the Challenging dataset. Fig. 2 shows the KL divergence between the Stein ICP and ground truth distribution using the point-to-point error metric as a function of the particle count. The right Y-axis is for the *Apartment* sequence only due to the difference of the scale. Overall the KL divergence for both translations (top) and rotations (bottom) improves with a larger number of particles. However, the improvement in KL divergence between 40 and 100 particles is minor compared to the change from 10 to 40 particles. In the subsequent experiments
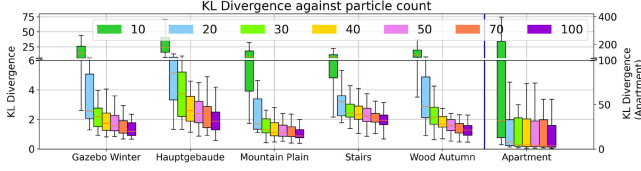
Fig. 2. KL Divergence against particle count. Overall, the best performance is achieved with 100 particles. Note however that there is little improvement in KL divergence between 40 and 100 particles.
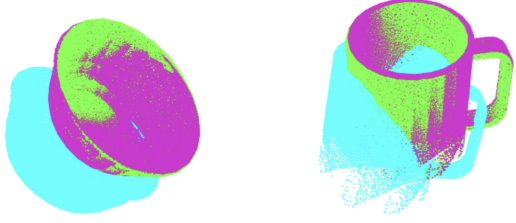

Fig. 3. Source (cyan), reference (magenta) and aligned cloud (green) of the *bowl* (left) and the *mug* (right). The *bowl* is rotationally symmetric around yaw, while handle of the *mug* constrains the yaw estimate.
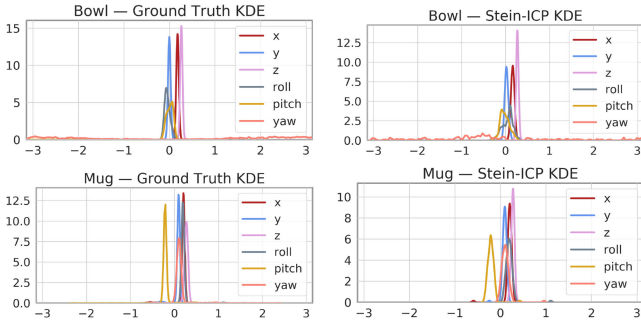

Fig. 4. Visualization of the pose distribution estimated by a kernel density estimator (KDE) for the ground truth and Stein ICP samples. The broad yaw distribution correctly captures the rotational symmetry of the round *bowl* (top). This is in contrast to the peaked yaw distribution of the *mug* (bottom) which has no symmetries.

we will use 100 particles as this reliably achieves high quality results.

B. Distribution Quality Analysis

We start with a qualitative evaluation of the distribution estimated by Stein ICP on two examples from the RGB-D dataset. In Fig. 3 we show the source (cyan), reference (magenta), and aligned (green) clouds for a *bowl* (left) and a *mug* (right). The kernel density estimates (KDE) of both the ground truth samples (left) and Stein ICP samples (right) are shown in Fig. 4. We can see how for the *bowl* (top), which has a rotational symmetry around yaw, the estimates are peaked with the exception of yaw which spans the full $[-\pi, \pi]$ range, both in the ground truth distribution and Stein ICP. By contrast the distributions for the *mug* (bottom) are peaked for all parameters. As the handle of the *mug* clearly identifies the single correct solution, this uni-modal distribution is what we would expect. These results demonstrate that the distributions recovered by Stein ICP match the ground truth distributions and that both distributions agree with our intuitions about object alignment ambiguity and uniqueness.

Sequence	Method	KL Div.	OVL
Apartment	Ours	$5.7\ [1.2, 4e^2]$	$0.7\ [0.4, 0.9]$
	Bayesian-multi	$4.2\ [0.8, 1e^2]$	$0.7\ [0.3, 0.9]$
	Bayesian ICP	$4.6\ [1.6, 2e^2]$	$0.7\ [0.4, 0.8]$
	COV-3D ICP	$2e^2\ [17.2, 2e^4]$	$0.3\ [0.1, 0.5]$
	Closed-form ICP	$1e^4\ [8e^2, 9e^6]$	$0.0\ [0.0, 0.0]$
Gazebo Winter	Ours	$1.1\ [0.5, 3e^2]$	$0.9\ [0.4, 0.9]$
	Bayesian-multi	$2.9\ [1.0, 4e^2]$	$0.8\ [0.4, 0.9]$
	Bayesian ICP	$3.0\ [1.2, 3e^2]$	$0.8\ [0.4, 0.9]$
	COV-3D ICP	$42.4\ [7.7, 1e^3]$	$0.3\ [0.1, 0.4]$
	Closed-form ICP	$1e^5\ [4e^4, 1e^7]$	$0.0\ [0.0, 0.0]$
Hauptgebaude	Ours	$0.6\ [0.3, 17.6]$	$0.9\ [0.8, 0.9]$
	Bayesian-multi	$1.9\ [1.3, 13.2]$	$0.8\ [0.7, 0.8]$
	Bayesian ICP	$1.8\ [1.3, 15.7]$	$0.8\ [0.7, 0.8]$
	COV-3D ICP	$2e^2\ [55.5, 8e^2]$	$0.2\ [0.1, 0.3]$
	Closed-form ICP	$4e^4\ [1e^4, 1e^5]$	$0.0\ [0.0, 0.0]$
Mountain Plain	Ours	$2.1\ [1.6, 12.7]$	$0.7\ [0.6, 0.9]$
	Bayesian-multi	$62.0\ [1.9, 5e^2]$	$0.5\ [0.3, 0.8]$
	Bayesian ICP	$88.9\ [4.3, 3e^2]$	$0.5\ [0.3, 0.7]$
	COV-3D ICP	$16.1\ [6.4, 69.4]$	$0.3\ [0.2, 0.5]$
	Closed-form ICP	$4e^4\ [1e^4, 1e^5]$	$0.0\ [0.0, 0.0]$
Stairs	Ours	$2.6\ [0.9, 3.7]$	$0.7\ [0.5, 0.8]$
	Bayesian-multi	$1.9\ [1.0, 5.7]$	$0.8\ [0.5, 0.9]$
	Bayesian ICP	$2.1\ [0.9, 4.9]$	$0.7\ [0.5, 0.9]$
	COV-3D ICP	$16.1\ [11.3, 1e^2]$	$0.3\ [0.2, 0.4]$
	Closed-form ICP	$4e^3\ [6e^2, 8e^4]$	$0.0\ [0.0, 0.0]$
Wood Autumn	Ours	$0.6\ [0.4, 1.7]$	$0.9\ [0.8, 0.9]$
	Bayesian-multi	$1.2\ [0.8, 2.5]$	$0.9\ [0.8, 0.9]$
	Bayesian ICP	$1.2\ [0.7, 3.2]$	$0.9\ [0.6, 0.9]$
	COV-3D ICP	$8.1\ [7.1, 9.4]$	$0.4\ [0.3, 0.5]$
	Closed-form ICP	$4e^4\ [2e^4, 1e^5]$	$0.0\ [0.0, 0.0]$

Next, we perform a quantitative comparison between our method and the state-of-the-art probabilistic ICP methods namely *Bayesian ICP* [34], a multi-chain Bayesian ICP–denoted by Bayesian-multi–this is the result of running 10 chains of Bayesian ICP and compounding the results, *COV-3D ICP* [32], and a popular benchmark *Closed-form ICP* [13] on the Challenging dataset. The results are summarised in Table I which shows the median of the KL divergence and OVL as well as the 10th and 90th quantile in the brackets. The divergence and OVL are computed over all consecutive pairs in each sequence with all methods using the point-to-plane error metric.

From those numbers we can see that our method outperforms both *COV-3D* and *Closed-form* ICP on all of the sequences. The large values of KL divergence and small values of OVL for *COV-3D* and *Closed-form* ICP indicate that these methods fail to provide correct uncertainty estimates. The *Closed-form* ICP method only provides reliable estimates in situations where uncertainty arises solely due to sensor noise [32], which is not the case in many of the scenes in the Challenging dataset. The *COV-3D* method relies on an accurate initial pose uncertainty estimation to provide a correct pose covariance. This causes the algorithm to under- or over-estimate the pose covariance when the initial covariance estimate is either inexact or not available [32].

When comparing Stein ICP with *Bayesian ICP* and *Bayesian-multi* we observe that in most scenes Stein ICP performs better. The only exception is *Apartment* where *Bayesian ICP* performs
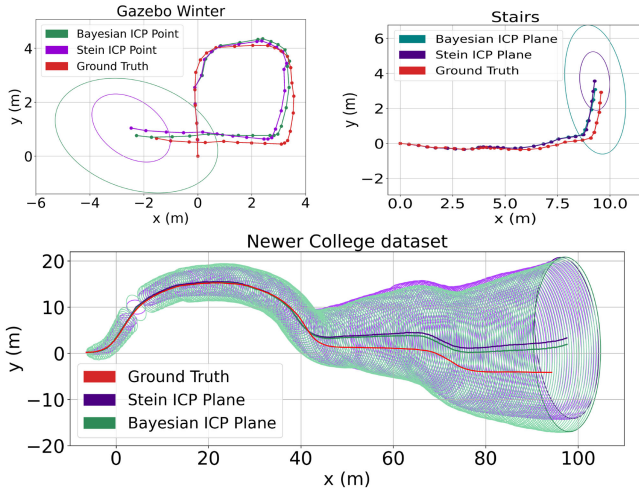
Fig. 5. Stein ICP and Bayesian ICP odometry overlaid on the ground truth trajectory for *Gazebo Winter*, *Stairs*, and Newer College dataset [45]. The ellipses represent the 95% confidence set of the compound covariance at the last pose of odometry for each method for sequences and along the entire trajectory for [45].
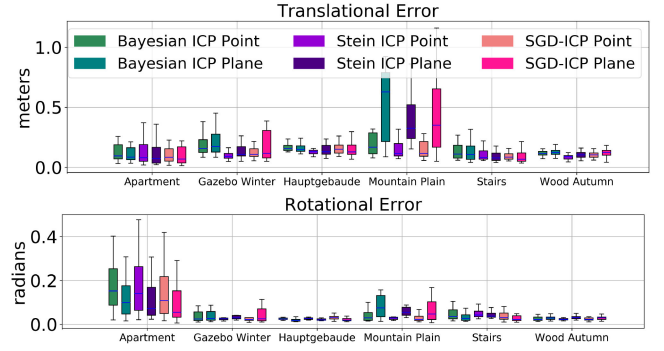

Fig. 6. Translational (top) and rotational (bottom) error distributions over entire trajectories of Stein ICP in comparison to Bayesian ICP and SGD-ICP method on the Challenging dataset. Overall, Stein ICP gives equivalent performance to that of SGD-ICP indicating that Stein ICP does not suffer from a reduction in quality despite providing uncertainty estimates.

better. As the KL divergence is computed based on the covariance of the samples, the fact that Stein ICP only produces 100 samples while Bayesian ICP produces 1000 samples can have an impact on the KL divergence. However, overall, both methods produce results of comparable quality. Similar results can be observed for the OVL measure, where Stein ICP performs better overall.

C. Odometry

In this section we use Stein ICP to compute odometry estimates by propagating the frame-to-frame expectations of the ICP transformation distributions to obtain a global pose estimate. These results are compared with the odometry estimate obtained by Bayesian ICP. As our method provides samples of the transformation parameters and can capture multi-modal distributions, the overall expectation and covariance may not represent the underlying correct pose.

For a sequence of l scans the individual ICP pose estimates are combined as $T_E = T_0^1 T_1^2 \ldots T_{l-1}^l$. Where T_{l-1}^l is a 4×4 homogeneous transformation matrix that captures the mean rotation and translation between scan l and $l-1$. A 4th order approximation [6] is used to combine the individual scan-to-scan covariances. The resulting trajectories for *Gazebo Winter*, *Stairs*, and Newer College dataset [45] are overlaid onto the ground plane to visualize the estimated trajectories in Fig. 5. The lines display the trajectories and the ellipses show the $3 - sigma$ (95%) confidence sets for uncertainty estimation of Stein ICP and Bayesian ICP.

The plots in Fig. 5 show that the mean trajectory obtained from Stein ICP and Bayesian ICP estimates are similar and track the ground truth trajectory well. The covariance ellipses estimated using Bayesian ICP are larger and less certain than those of Stein ICP. As the odometry estimates are obtained by accumulating transformations small differences in errors can add up over the trajectory, amplifying the small differences between Stein ICP and Bayesian ICP reported in Table I. Overall both methods provide good odometry trajectories and covariance

estimates. The ability to estimate odometry trajectories including uncertainty is crucial in many robotic applications such as autonomous navigation and SLAM. More generally any method reliant on pose estimation can use the uncertainty provided by these methods for decision making.

D. Impact of Distribution Estimation on Mean Pose Estimation Quality

Estimating the full distribution over pose parameters is useful. However, if a single solution exists what, if any, is the price in pose estimation accuracy being paid in comparison to a standard point estimation method? To answer this question we examine the trajectory estimation errors for Stein ICP, Bayesian ICP, and SGD-ICP all using both the point-to-point and the point-to-plane error metric. To measure the local accuracy of the trajectory estimation we compute the *relative pose error* [51] (RPE) for the SGD-ICP solution and each sample of the distributions produced by Stein ICP and Bayesian ICP. The RPE is defined as follows:

$$E = (\hat{T}_i^{-1}\hat{T}_{i+\Delta})^{-1}(T_i^{-1}T_{i+\Delta}), \qquad (18)$$

where $(\hat{T}_i^{-1}\hat{T}_{i+\Delta})$ is the ground truth transformation and $(T_i^{-1}T_{i+\Delta})$ is the ICP transformation between scan i and $i+\Delta$, with $\Delta = 1$ here.

The translational error is computed using the Euclidean distance of the translation components of E while the rotational errors are computed using absolute angular differences between the orientation of the ground truth and the estimated pose. For sample based methods we obtain a single error by averaging together the per sample errors. In Fig. 6 we show the translational (top) and rotational (bottom) error distributions over all the consecutive scan pairs of each used sequence independently.

From this we can see that the estimates of the distribution based methods are not any worse than those of a point estimation method. The choice of error metric has a bigger impact on the results than whether or not a distribution is estimated. This means that we do no give up any accuracy as far as transformation estimation goes with Stein ICP.

E. Runtime

Finally, we compare the run-time of Stein ICP, Bayesian ICP and Bayesian-multi. Traditionally, obtaining uncertainty
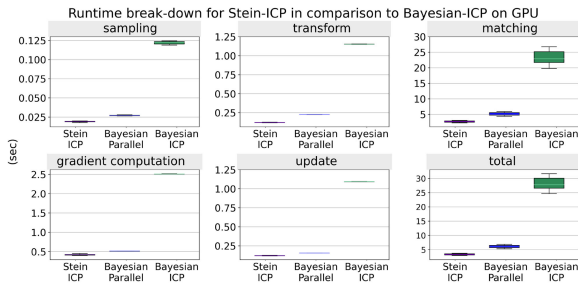
Fig. 7. Break-down of the runtime (sec) on the GPU. The times correspond to mini-batch sampling (top left), transforming point cloud with the updated transformation time (top middle), matching time (top right), gradient time (bottom right), and update time (bottom middle). Bottom right shows the total time. For Stein ICP, the gradients computational time includes the kernel estimation and repulsive force. These results demonstrate that Stein ICP is more efficient than both versions of Bayesian ICP.

estimates for ICP methods was computationally expensive, especially if accurate but slow Markov Chain Monte Carlo based methods were used. Runtime efficiency is one of the main advantages of Stein ICP due to its parallelism and leverage of GPUs. In the following, we present a break down of the runtime by the individual components of Stein ICP compared to Bayesian ICP. To this end, we use the *Gazebo Winter* dataset and run Stein ICP for 100 iterations using 100 particles. In Bayesian ICP, each iteration provides us with a sample. However we need to run the method over a certain number of iterations before the Markov Chain becomes stable; a process known as burn-in. To this end, we run Bayesian ICP for 1100 iterations to collect 1000 sequential samples after discarding initial 100 burn-in samples. We also run 10 parallel chains of Bayesian-multi where each chain collects 100 samples with a burn-in period of 100 iterations to produce a distribution of equivalent quality. All computations are performed on a desktop computer with an NVidia Titan-V GPU.

Fig. 7 presents the runtime of the various components along the Y-Axis. Even though Stein ICP is run with 100 particles, which is roughly equivalent to running 100 independent SGD-ICP instances, Stein ICP takes $\frac{1}{8^{th}}$ of the time Bayesian ICP requires for each ICP component. This is because Stein ICP's particles can trivially be parallelized on a GPU. In contrast, Bayesian ICP produces samples under the Markov assumption, *i.e.* every sample depends on the previous one, therefore it cannot reliably exploit GPU parallelism to generate samples. Even parallel Bayesian ICP (if run for efficiency) consumes relatively more time. This is due to the burn-in period necessary for each parallel chain of Bayesian ICP to ensure convergence to the true distribution. Stein ICP ability to perform computations in parallel on a per particle basis means that the GPU implementation can do more work for 100 particles in roughly the same time as required by a single particle. This results in over 5 times speedup compared to Bayesian ICP. Both Cov-3D and Closed-form methods require at least solving a point-to-plane ICP first (2.62 ± 0.19 sec on the GPU) in addition to the time required to compute the covariances. In addition, Cov-3D requires 12 registration steps for the unscented transform.

VI. CONCLUSION

In this letter we devised Stein ICP, a probabilistic generalization of the popular ICP that provides accurate posterior pose distributions. Stein ICP exploits the stochastic optimization

nature of SGD-ICP combining it with the recent Stein variational gradient descent (SVGD) algorithm to approximate posterior distributions of transformation parameters between two point clouds using particles. The independent nature of these particles makes Stein ICP suitable for GPU parallelization. Extensive experiments using both RGB-D as well as LiDAR data demonstrate the capability of our method in providing high-quality pose parameter distributions in only a few seconds of computation. In contrast to Markov chain Monte Carlo solutions to probabilistic point-cloud matching, Stein ICP propagates gradients of particles in parallel, connecting gradient descent on the objective function with a repulsive term that ensures particles are aligned and capture the tail of the underlying distribution. This opens up new possibilities to leverage calibrated uncertainty estimation in applications requiring robust point cloud matching such as simultaneous localization and mapping or 3D reconstruction. Another exciting direction on using the uncertainty provided by the method is for grasping unknown objects based on a list of known objects with associated grasping positions. The uncertainty can be used to indicate which of the known objects is a better match and which direction (with less uncertainty) to grasp the unknown object.

REFERENCES

[1] F. A. Maken, F. Ramos, and L. Ott, "Speeding up iterative closest point using stochastic gradient descent," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2019, pp. 6395–6401.

[2] Y. Aoki, H. Goforth, R.A. Srivatsan, and S. Lucey, "Pointnetlk: Robust and efficient point cloud registration using pointnet," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7163–7172.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep.1987.

[4] M. Barczyk and S. Bonnabel, "Towards realistic covariance estimation of ICP-based Kinect V1 scan matching: The 1D case," in *Proc. Amer. Control Conf.*, 2017, pp. 4833–483.

[5] M. Barczyk, S. Bonnabel, J. E. Deschaud, and F. Goulette, "Invariant EKF design for scan matching-aided localization," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 6, pp. 2440–2448, Nov. 2015.

[6] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Trans. Robot.*, vol. 30, no. 3, pp 679–693, Jun. 2014.

[7] O. Bengtsson and A. J. Baerveldt, "Robot localization based on scan-matching-estimating the covariance matrix for the IDC algorithm," *Robot. Auton. Syst.*, vol. 44, no. 1, pp 29–40, 2003.

[8] P. Besl and N. D. McKay, "Method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[9] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2743–2748.

[10] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg, "Globally consistent 3D mapping with scan matching," *Robot. Auton. Syst.*, vol. 56, no. 2, pp 130–142, 2008.

[11] M. Bosse and R. Zlot, "Map matching and data association for large-scale two-dimensional laser scan-based SLAM," *Int. J. Robot. Res.*, vol. 27, no. 6, pp. 667–691, 2008.

[12] V. Caglioti, F. Mainardi, M. Pilu, and D. Sorrenti, "Improving pose estimation using image, sensor and model uncertainty," in *Proc. Brit. Mach. Vis. Conf.*, 1994, pp. 1–10.

[13] A. Censi, "An accurate closed-form estimate of ICP's covariance," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3167–3172.

[14] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image Vis. Comput.*, vol. 10, no. 3, pp 145–155, 1992.

[15] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Proc. Int. Conf. Pattern Recognit.*, 2002, pp. 545–548.

[16] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 1556–1563.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., Ser. B*, vol. 39, no. 1, pp 1–38, 1977.

[18] B. Eckart, K. Kim, and J. Kautz, "HGMR: Hierarchical gaussian mixtures for adaptive 3D registration," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2018, pp. 705–721.

[19] S. Izadi *et al.*, "Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2011, pp. 559–568.

[20] A. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, pp 1145–1153, 2003.

[21] W. Gao and R. Tedrake, "Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11095–11104.

[22] B. K. P.Horn, "Closed-form solution of absolute orientation using unit quaternions," *J. Opt. Soc. Amer. A*, vol. 4, no. 4, pp. 629–42, 1987.

[23] F. H.Inman and E. L. Bradley, "The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities," *Commun. Statist.-Theory Methods*, vol. 18, pp 3851–3874, 1989.

[24] T. M. Iversen, A. G. Buch, and D. Kraft, "Prediction of ICP pose uncertainties using monte carlo simulation with synthetic depth images," in *Proc. Int. Conf. Intell. Robots Syst.*, 2017, pp. 4640–4647.

[25] S. Julier, J. Uhlmann, and H. F. Durrant-Whyte, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 477–482, Mar. 2000.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc Int. Conf. Learn. Representations*, 2015, pp. 1–13.

[27] A. Der Kiureghian and O. Ditlevsen "Aleatory or epistemic? Does it matter?," *Struct. Saf.*, vol. 31, no. 2, pp. 105–12, 2009.

[28] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1817–1824.

[29] D. Landry, F. Pomerleau, and P. Giguére, "CELLO-3D: Estimating the covariance of ICP in the real world," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 8190–8196.

[30] Q. Liu and D. Wang, "Stein variational gradient descent: A general purpose bayesian inference algorithm," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 2378–2386.

[31] J. Luck, C. Little, and W. Hoff, "Registration of range data using a hybrid simulated annealing and iterative closest point algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 3739–3744.

[32] A. Barrau, M. Brossard, and S. Bonnabel, "A new approach to 3D ICP covariance estimation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 744–751, Apr. 2020.

[33] B. Ma and R. Ellis, "Surface-based registration with a particle filter," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Intervention*, 2004, pp. 566–573.

[34] F. A. Maken, F. Ramos, and L. Ott, "Estimating motion uncertainty with bayesian ICP," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8602–8608.

[35] T. Masuda, "Generation of geometric model by registration and integration of multiple range images," in *Proc. Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 254–261.

[36] T. Masuda and N. Yokoya, ''Robust method for registration and segmentation of multiple range images," *Comput. Vis. Image Understanding*, vol. 61, no. 3, pp. 295–307, 1995.

[37] E. Mendes,P. Koch, and S. Lacroix, "ICP-based pose-graph SLAM," *IEEE Int. Symp. Saf., Secur., Rescue Robot.*, 2016, pp. 195–200.

[38] W. Ming-Long, "An augmented reality system using improved-iterative closest point algorithm for on-patient medical image visualization," *Sensors*, vol. 18, no. 8, 2018, Art. no. 2505.

[39] R. M. Neal, *Bayesian Learning for Neural Networks*. Berlin, Heidelberg, Germany: Springer-Verlag, 1996.

[40] R. M. Neal, "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, vol. 54, 2010, pp. 113–162.

[41] I. K. Park, M. Germann, M D. Breitenstein, and H. Pfister, "Fast and automatic object pose estimation for range images on the GPU," *Mach. Vis. Appl.*, vol. 21, no. 5, pp. 749–766, Aug. 2010.

[42] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," *Found. Trends Robot.*, vol. 4, no. 1, pp. 1–104, May 2015.

[43] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart, "Challenging data sets for point cloud registration algorithms," *Int. J. Robot. Res.*, vol. 31, pp. 1705–1711, 2012.

[44] S. M. Prakhya, L. Bingbing, Y. Rui, and W. Lin, "Covariance of ICP Algorithm A. Closed-form Estimate of 3D ICP Covariance," in *Proc. Int. Conf. Mach. Vis. Appl.*, 2015, pp. 526–529.

[45] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The newer college dataset: Handheld LiDAR, inertial and vision with ground truth," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4353–4360.

[46] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp 400–407, Sep. 1951.

[47] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard, "On the position accuracy of mobile robot localization based on particle filters combined with scan matching," in *Proc. Int. Conf. Intell. Robots Syst.*, 2012, pp. 3158–3164.

[48] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.

[49] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," *Robot.: Sci. Syst.*, vol. 2, no. 4, p. 435, 2009.

[50] S. Shi, Z. You, K. Zhao, Z. Wang, C. Ouyang, and Y. Cao, "A. 6-DOF navigation method based on iterative closest imaging point algorithm," *Sci. Rep.*, vol. 7, no. 1, pp. 1–1, 2017.

[51] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[52] J. M. Tur, "Onto computing the uncertainty for the odometry pose estimate of a mobile robot," in *Proc. IEEE Conf. Emerg. Technol. Factory Automat.*, 2007, pp. 1340–1345.

[53] D. Wang, Z. Tang, C. Bajaj, and Q. Liu, "Stein variational gradient descent with matrix-valued Kernels," in *Proc. Neural Inf. Process. Syst.*, 2019, Art. no. 7834.

[54] Y. Wang and J. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3523–3532.

[55] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 681–688.

[56] C. Zhang, J. Butepage, H. Kjellstrom, and S. Mandt, "Advances in variational inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 2008–2026, Aug. 2019.