Estimating Motion Uncertainty with Bayesian ICP

Fahira Afzal Maken[1,*], Fabio Ramos[1,2] and Lionel Ott[1]

Abstract— Accurate uncertainty estimation associated with the pose transformation between two 3D point clouds is critical for autonomous navigation, grasping, and data fusion. Iterative closest point (ICP) is widely used to estimate the transformation between point cloud pairs by iteratively performing data association and motion estimation. Despite its success and popularity, ICP is effectively a deterministic algorithm, and attempts to reformulate it in a probabilistic manner generally do not capture all sources of uncertainty, such as data association errors and sensor noise. This leads to overconfident transformation estimates, potentially compromising the robustness of systems relying on them. In this paper we propose a novel method to estimate pose uncertainty in ICP with a Markov Chain Monte Carlo (MCMC) algorithm. Our method combines recent developments in optimization for scalable Bayesian sampling such as stochastic gradient Langevin dynamics (SGLD) to infer a full posterior distribution of the pose transformation between two point clouds. We evaluate our method, called Bayesian ICP, in experiments using 3D Kinect data demonstrating that our method is capable of both quickly and accurately estimating pose uncertainty, taking into account data association uncertainty as reflected by the shape of the objects.

I. INTRODUCTION

Iterative closest point (ICP) [3] is a widely used algorithm in robotics and computer vision to register two point clouds by iteratively minimizing the alignment error between them. ICP takes two input point clouds, source and reference, and an initial transformation estimate. The alignment error is minimized by repeatedly finding sets of corresponding point pairs in the two point clouds and using these to estimate a transformation that minimizes the distance between matching points. A large number of ICP variants have been proposed in the literature, building on the initial formulation proposed in [3]. Many methods differ in the cost-function used, e.g. point-to-point [3], point-to-plane [8], and plane-to-plane [20] to name a few. These and many other variants were extensively evaluated in [16], [19].

There are several potential sources of error and uncertainty in the pose estimate obtained by ICP. For example, sensor noise and the presence of local minima in the cost function. Another source of uncertainty comes from ambiguities in the point cloud structure, such as long featureless corridors or symmetries in objects like bowls or water bottles. This results in erroneous data associations which can cause significant estimation errors and make them impractical for localization or mapping tasks.

Current methods that estimate ICP pose uncertainty are either offline due to their high-computational cost or produce overconfident uncertainty estimates which can lead to navigation failure [10], [2], [7]. The reason for overconfident estimates stems from the fact that point re-associations are not taken into account in the pose uncertainty estimation step, making them unsuitable for point-to-point variants of ICP [5].

A principled way to incorporate uncertainty into the ICP pose estimation process is to use a Bayesian formulation in which a prior distribution and a likelihood function are used to obtain a posterior distribution given observations. Unfortunately analytical solutions to such a Bayesian formulation are typically intractable and approximate methods have to be employed. One such approximation technique is Markov Chain Monte Carlo (MCMC) which can produce a sample-based approximation to the posterior [18]. However, traditional MCMC algorithms are still computationally demanding as they need to consider the entire dataset to produce a single sample.

In this paper we use a scalable class of MCMC techniques, called stochastic gradient Langevin dynamics (SGLD), which combines stochastic optimization with Langevin dynamics [13], [14], [15], [24]. SGLD enables any stochastic gradient descent (SGD) based optimization to produce samples from the true posterior by injecting a controlled amount of noise into the optimization. We combine this idea with SGD-ICP [1], an ICP method that uses SGD optimization, to obtain a method which produces a pose distribution by generating samples from the posterior distribution.

The main contribution of this paper is the introduction of Bayesian ICP, an efficient and scalable ICP variant that provides a high-quality approximation over the posterior transformation. This distribution encodes both the expected pose transformation, as provided by other ICP methods, as well as uncertainty information. A C++ implementation of Bayesian ICP is available at `https://bitbucket.org/fafz/bayesian-icp`.

II. RELATED WORK

Various approaches have been proposed in the literature to estimate the pose uncertainty of ICP using closed

* Corresponding author: fafz3958@uni.sydney.edu.au
[1] School of Computer Science, The University of Sydney, Australia
[2] NVIDIA, USA

form solutions. In [9], it is assumed that there exists a perfect one-to-one correspondence between the source and reference cloud. Under this assumption isotropic Gaussian noise is used to model the uncertainty in the position of points. It is shown that rotation errors are distributed according to a 4-dimensional Bingham distribution. Another way to measure uncertainty in the pose estimate relies on the Hessian of the cost function. These closed form solutions are based on the underlying assumption that the cost function of ICP can be linearized around the point of convergence. This assumption makes it possible to use a least square estimate [11], [25] to derive the Hessian's covariance matrix analytically [2], [6], or otherwise it can be approximated numerically by sampling [4]. This type of uncertainty estimation only accounts for errors which are caused by the structure of the environment but not those due to sensor noise.

A method tackling sensor noise is proposed in [7], which computes a sensor's noise error covariance as a function of an error metric in the 2D case. An extension of this idea for the 3D case is proposed in [17]. This approach makes certain assumptions which may restrict their use, e.g. it is assumed that the algorithm converges to the attraction region of the true solution and that data re-association in that region does not significantly impact the objective function.

Another approach to estimate the ICP pose uncertainty is the use of sampling based methods, such as Monte Carlo algorithms [2], [10]. These methods sample a large number of ICP pose transforms, requiring a complete ICP solution each time, and use the covariance of these samples as a measure of uncertainty. If a model of the environment is available, such as a computer aided design (CAD) drawing, then sensor noise can be taken into account by generating multiple scans of the environment using a sensor model [10]. While such methods are capable of providing accurate estimates of the ICP pose uncertainty, their high computational cost makes them often unsuitable for online robotics applications. Additionally, the need for an environment model to account for sensor noise is a further limiting factor.

In this paper we propose a novel ICP method which estimates the pose distributions using stochastic gradient Langevin dynamics (SGLD). In comparison to standard MCMC based methods our method is highly efficient, joining SGD-ICP [1] and SGLD to overcome the computational costs of typical MCMC approaches. In contrast to analytical methods information about the Hessian is not needed nor are assumptions made about the consistency of data associations. The general sampling based nature of the proposed approach makes the approach amenable to be used with many existing ICP variants.

III. BACKGROUND

In the following, we first describe the standard ICP algorithm, before giving an overview of stochastic gradient descent ICP (SGD-ICP) [1], which forms the basis of our proposed Bayesian ICP method.

A. Standard ICP

Given two input point clouds, $S = \{s_i\}_{i=1}^N$ and $R_{ef} = \{r_i\}_{i=1}^M$, where $s_i, r_i \in \mathbb{R}^3$, and an initial estimate of the transformation parameters $\theta = \{x, y, z, \text{roll}, \text{pitch}, \text{yaw}\}$, the standard ICP formulation performs a rigid registration between the clouds in two steps: First, correspondences between point pairs in the two clouds are found on the basis of the Euclidean distance using the current transformation θ. Second, a rigid transformation between the corresponding point pairs is found that minimizes a cost function. These two steps are repeated until a convergence criteria, such as error threshold or iteration number, is met.

The loss $\mathcal{L}$ optimised by ICP in the second step is a function of θ and captures how well the transformation aligns the point pairs, i.e.:

$$\underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta) = \frac{1}{N} \sum_i^N ||(R\, s_i + u) - r_i||^2, \qquad (1)$$

where $u \in \mathbb{R}^{3\times 1}$ is a translation vector and $R \in \mathbb{R}^{3\times 3}$ is a rotation matrix.

B. Stochastic Gradient Descent ICP

SGD-ICP [1] employs stochastic gradient descent (SGD) to solve the ICP optimization problem (1). In each iteration t a mini-batch $\mathcal{M}_t$ of size m is sampled from the source cloud S to which the corresponding points in R_{ef} are selected using standard ICP data association methods. The average gradients $\bar{g}$ of the parameter vector θ with respect to the loss $\mathcal{L}$ are computed as follows:

$$\bar{g}(\theta_t^{1:3}, \mathcal{M}_t) = \frac{1}{m} \sum_i^m \left((R_t\, s_i + u_t) - r_i\right) \frac{\partial u_t}{\partial \theta_t^{1:3}}, \qquad (2)$$

$$\bar{g}(\theta_t^{4:6}, \mathcal{M}_t) = \frac{1}{m} \sum_i^m \left((R_t\, s_i + u_t) - r_i\right) \frac{\partial R_t}{\partial \theta_t^{4:6}}\, s_i, \qquad (3)$$

where $\bar{g}(\theta_t^{1:3}, \mathcal{M}_t)$ are the gradients of the translation components and $\bar{g}(\theta_t^{4:6}, \mathcal{M}_t)$ are the rotational gradients. Using this gradient information the estimate of the transformation parameters θ can be updated iteratively with the following update rule:

$$\theta_{t+1} = \theta_t - \alpha A\bar{g}(\theta_t, \mathcal{M}_t), \qquad (4)$$

where the matrix $A \in \mathbb{R}^{6\times 6}$ acts as a pre-conditioner and the learning rate α dictates how quickly parameter values change.

By using small mini-batches, instead of the full point cloud, in each iteration SGD-ICP is computationally much more efficient than standard ICP methods. Importantly, however, the quality of the estimated transformation is equal to that of the full batch ICP approach [1].

IV. BAYESIAN ICP

In the following we derive our proposed method Bayesian ICP, combining ideas from SGD-ICP [1] and SGLD [24] to obtain an efficient ICP method capable of estimating a pose distribution. As we will see the mini-batch based formulation of the two algorithms allows for an elegant combination of the two.

SGLD [24] shows that adding the right amount of noise to a standard stochastic gradient descent based optimisation results in iterations converging to samples from the true posterior distribution, thus enabling Bayesian inference. As SGD-ICP is based on SGD, it is a good candidate for applying the ideas of SGLD to obtain a distribution over the transformation parameters θ. To this end the general SGD-ICP update rule (4) is modified by adding a prior $p(\theta)$ over the transformation parameters θ and injecting Gaussian noise $\eta_t \sim \mathcal{N}(0, A\alpha)$ at each iteration. With these additions we obtain the following general form for Bayesian ICP:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{2} A\Big(-\nabla \log p(\theta_t) + N\bar{g}(\theta_t, \mathcal{M}_t)\Big) + \eta_t, \quad (5)$$

where N is the size of the dataset, and $\nabla \log p(\theta_t)$ is the gradient of the log of the prior. SGLD maintains the balance between the amount of injected noise and the gradient step sizes in order to match the variance of the samples to the variance of the posterior. The convergence rates of this process have been explored in detail in [21].

The goal in ICP is to estimate a 6 DoF transformation which consists of three translations $\theta^{1:3}$ and three rotations $\theta^{4:6}$. The priors on these are modelled using Gaussian distributions for the translations and von Mises [23] distributions for the rotations. Disregarding the normalizing constant term, the gradients of the log of Gaussian and von Mises prior distribution are:

$$-\nabla_\theta \log p(\theta^{1:3}) = (\theta^{1:3} - \mu^{1:3})/\sigma^{1:3}, \quad (6)$$

$$-\nabla_\theta \log p(\theta^{4:6}) = \kappa^{4:6} \sin(\theta_t^{4:6} - \mu^{4:6}), \quad (7)$$

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

Combining these priors with the SGD-ICP update rule (4), and using translational and rotational gradients (2) and (3) in the SGLD update equation (5) we arrive at:

$$\theta_{t+1}^{1:3} = \theta_t^{1:3} - \frac{\alpha}{2} A\Bigg[(\theta_t^{1:3} - \mu^{1:3})/\sigma^{1:3} + \frac{N}{m} \sum_i^m \Big((R_t\, s_i + u_t) - r_i\Big)\frac{\partial u_t}{\partial \theta_t^{1:3}}\Bigg] + \eta_t^{1:3} \quad (8)$$

for the translation parameters and

$$\theta_{t+1}^{4:6} = \theta_t^{4:6} - \frac{\alpha}{2} A\Bigg[\kappa^{4:6} \sin(\theta_t^{4:6} - \mu^{4:6}) + \frac{N}{m} \sum_i^m \Big((R_t\, s_i + u_t) - r_i\Big)\frac{\partial R_t}{\partial \theta_t^{4:6}}\, s_i\Bigg] + \eta_t^{4:6} \quad (9)$$

for the rotation parameters. As Bayesian ICP produces a collection of samples, as opposed to a single parameter vector, we can recover the posterior expectation $\mathbb{E}[\theta]$ of θ as an average over all T samples, i.e. $\mathbb{E}[\theta] = \frac{1}{T}\sum_{t=1}^{T}\theta_t$.

When using a fixed step size, SGLD can exhibit slow mixing of the distribution, especially when parameters have locally different curvatures. One approach to overcome this is preconditioned SGLD [13] which constructs a local transform that equalizes the gradients, similar to RMSProp [22]. In this way a simple fixed step size is adequate to obtain good convergence. This gradient preconditioning is incorporated via the matrix A and is computed as follows:

$$V(\theta_t) = \beta V(\theta_{t-1}) + (1-\beta)\bar{g}(\theta_t, \mathcal{M}_t) \odot \bar{g}(\theta_t, \mathcal{M}_t), \quad (10)$$

$$A(\theta_t) = diag\left(1 \oslash \left(\lambda 1 + \sqrt{V(\theta_t)}\right)\right), \quad (11)$$

where operators $\odot$ and $\oslash$ represent element-wise matrix product and division respectively. The hyper parameter λ controls the extremes of the curvature in the preconditioner while $\beta \in [0,1]$ balances the weights of the historical and current gradients.

Algorithm 1 outlines the steps performed by Bayesian ICP. First, a new mini-batch is obtained from the source cloud in line 2. This mini-batch is then transformed with the latest transformation matrix in line 3, before in lines 4 to 8 the corresponding point pairs between the mini-batch and the reference cloud are found and stored. Once corresponding pairs have been obtained, the stochastic gradient, preconditioning, and noise are computed in lines 9 to 12. With these quantities the translation and rotation parameters are updated in line 13 and 14 respectively. Finally, after T iterations the final set of parameter samples are returned in line 16.

V. EVALUATION

In this section we investigate the performance of Bayesian ICP in terms of the estimated pose distribution quality, impact of the number of samples, burn in length, run-time, as well as final pose estimation accuracy.

Experiments are performed on the RGB-D Scenes Dataset [12] which contains both partially occluded as well as a complete scans of household objects. The objects used from the dataset fall into two categories: under constrained and constrained. Under constrained objects have one or more symmetries, such as soda cans and bowls, while constrained objects have no symmetries, e.g. a coffee mug or a table. All of these objects are transformed to a known offset which is used as a ground truth pose. This dataset is selected as we expect obvious uncertainty, along the symmetry, for the under constrained objects and none for the constrained objects.

In order to properly evaluate the quality of the estimated pose distribution we require a baseline method. For this we obtain 10000 transformations by running the most commonly used standard ICP to completion, each time with a different initial transformation parameters

8604

Algorithm 1: Bayesian ICP

Input : Source $S = \{s_i\}$ and reference $R_{ef} = \{r_i\}$ clouds, Initial transformation parameters: θ_1, Source cloud size: N, Mini-batch size: m, Step size: α, $\lambda = 1 \times 10^{-8}$, $\beta = 0.9$, $V(\theta_0) \leftarrow 0$

Output: Transformation samples: $\{\theta_t\}_{t=1:T}$

1 **for** $t \leftarrow 1 : T$ **do**

2    $\mathcal{M}_t \leftarrow$ pick a mini-batch cloud of size m from S

3    $\mathcal{M}_t' \leftarrow$ transform mini-batch with θ_t

4   Pairs $\leftarrow \emptyset$

5   **for** $s_i' \in \mathcal{M}_t'$ **do**

6      $r_i \leftarrow$ closest point in R_{ef} to s_i'

7   Pairs $\leftarrow$ Pairs $\cup \quad \{s_i', r_i\}$

8   **end**

    // Estimate mean gradients

9    $\bar{g}(\theta_t; \mathcal{M}_t) \leftarrow \frac{1}{m} \sum_{s_i', r_i \in \text{Pairs}} (s_i' - r_i) \frac{\partial(s_i' - r_i)}{\partial \theta_t}$

    // Compute preconditioner and noise

10    $V(\theta_t) \leftarrow \beta V(\theta_{t-1}) + (1-\beta)\bar{g}(\theta_t; \mathcal{M}_t)\bar{g}(\theta_t; \mathcal{M}_t)$

11    $A(\theta_t) \leftarrow \left(1 \div \left(\lambda + \sqrt{V(\theta_t)} \right) \right)$

12    $\eta_t \leftarrow \mathcal{N}(0, \alpha A(\theta_t))$

    // Update parameters

13    $\theta_{t+1}^{1:3} \leftarrow \theta_t^{1:3} - 0.5\alpha A(\theta_t^{1:3}) \Big((\theta_t^{1:3} - \mu^{1:3})/\sigma^{1:3} + N\bar{g}(\theta_t^{1:3}; \mathcal{M}_t) \Big) + \eta_t^{1:3}$

14    $\theta_{t+1}^{4:6} \leftarrow \theta_t^{4:6} - 0.5\alpha A(\theta_t^{4:6}) \Big(\kappa^{4:6} \sin(\theta_t^{4:6} - \mu^{4:6}) + N\bar{g}(\theta_t^{4:6}; \mathcal{M}_t) \Big) + \eta_t^{4:6}$

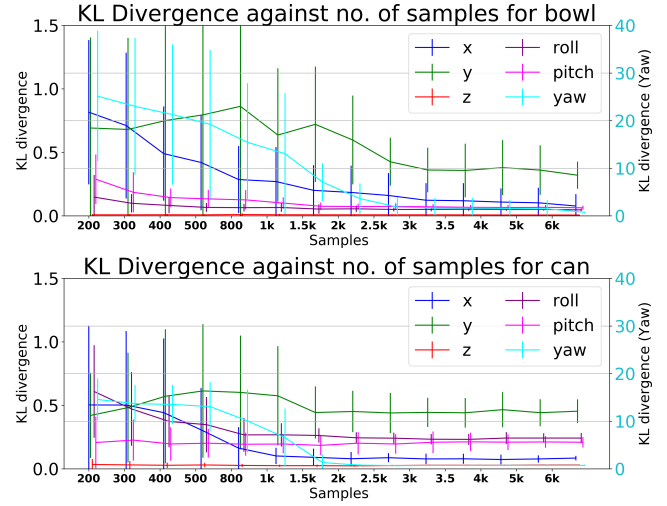15 **end**

16 **return** $\{\theta_t\}_{t=1:T}$

Fig. 1: KL divergence of Bayesian ICP compared to the baseline for two objects bowl (top) and can (bottom), as a function of the number of samples. The majority of the improvements occur during the first 2000 samples.

can be used. These values were coarsely picked by manual tuning. The prior mean and variance are set 0.0 and 0.125 respectively, however, when deployed on a platform these values could be obtained from the system's state estimation. All experiments were performed on a PC with an Intel Core i7-7700 CPU with 16GB of RAM. The algorithm was multi-threaded using OpenMP.

A. Sample count

We begin by evaluating the effect the number of samples has on the quality of the pose distributions estimated by Bayesian ICP. We allow Bayesian ICP to collect 6000 samples and repeat this ten times. Fig. 1 shows the evolution of mean and standard deviation of the KL divergence as the number of samples increases. The Y axis shows the KL divergence compared to the baseline, a second Y axis (right side) is used for yaw. Overall, increasing the sample size improves the quality of the distribution as both mean and standard deviation shrink. Though after a certain number of samples the iteration to iteration gains decrease. Based on these plots and other experiments 2000 to 3000 samples provide almost optimal results, and going forward we use 2000 samples in all other experiments.

B. Burn-in length

Burn-in, which discards samples from the beginning of a sampling sequence, is common in MCMC methods and ensures that the samples produced by the scheme come from the true underlying distribution. To analyze the impact the amount of points discarded by the burn-in process has on the final distribution quality, we use the samples collected in the previous section. However, we remove a variable number of initial points and then use the subsequent 2000 points to obtain a distribution estimate. In Fig. 2 we show the resulting KL divergence

drawn from $\pm 1\,\text{m}$ and $\pm 1\,\text{rad}$ for translation and rotation respectively. This resulting collection of ICP transformations acts as our baseline ground truth pose distribution. We additionally provide comparisons against a widely used online method [7] (Online-ICP) using the point-to-plane based implementation provided by [17].

To quantify the quality of the distributions we use the Kullback-Leibler (KL) divergence between the baseline distribution and the comparison method. The final transformation quality is measured as the Euclidean distance between the estimate pose and ground truth pose for the translation, and sum of absolute angular differences for the rotation.

In the experiments our method uses a step size of 0.0001 with a mini-batch size of 160 points. To aid with convergence and regulate SGLD noise, a block decay strategy is used for the step size which is decreased by 20% every 10 iterations towards the end of sampling. The parameter N in (5), which represents the size of the dataset, scales the mean gradient and thus impacts the choice of step size α. As the size of point clouds varies greatly, N is fixed to $N = 6000$ such that a single step size
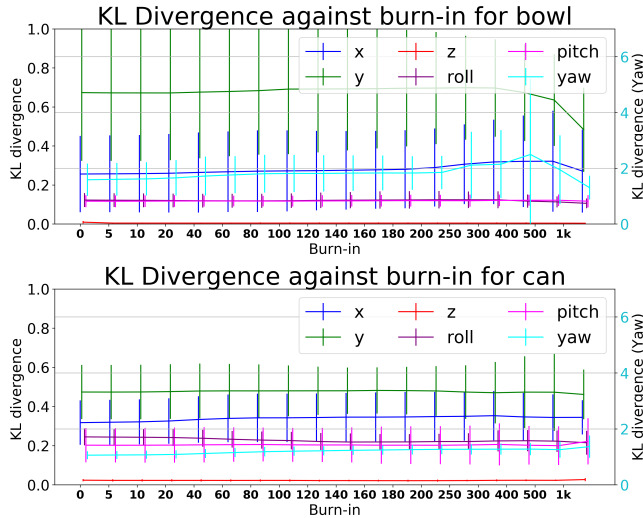
Fig. 2: KL divergence of Bayesian ICP compared to the baseline for two objects bowl (top) and can (bottom), as a function of burn-in samples. Overall burn-in does not have any impact on the quality for small initial true transform.

when compared with the baseline distribution. As we can see the size of the burn-in has no visible effect on the distribution quality. This can be explained by the fact that in the case of ICP the initial guess is close to the distribution of possible solutions and as such the sampling scheme very quickly obtains samples from the true distribution. However, for large discrepancies between initial and final transformation a certain amount of burn-in might be beneficial.

C. Distribution estimation quality

Fig. 3 shows a comparison of kernel density estimation (KDE) models of the baseline distribution (left) and the proposed Bayesian ICP method (right) on three objects. The X axis shows the transformation parameter values while the Y axis shows the density. In Fig. 4 the corresponding source scan (cyan), reference scan (magenta) and Bayesian-ICP aligned scans (green) are shown, highlighting the geometry of each object and the accuracy of the final pose. In every instances all parameters except for yaw converge close to the mean of the ground truth pose. For yaw the ground truth value is not obtained for under-constrained objects, as these are symmetric in yaw and as such all values result in the same solution. While the distribution in yaw is to some extent visually different between the baseline and Bayesian ICP solution both capture the overall picture of uncertainty. For constrained objects, which have no symmetries, the distributions show a single peak near the ground truth value for all parameters.

We provide numerical comparisons in Table I between our proposed method (Bayesian ICP) and a commonly used method [7] (Online-ICP) which provides a $\mathbb{R}^{6\times6}$ covariance matrix. Bayesian ICP is run ten times, obtaining 2000 samples each time. The table shows the mean

(a) Bowl

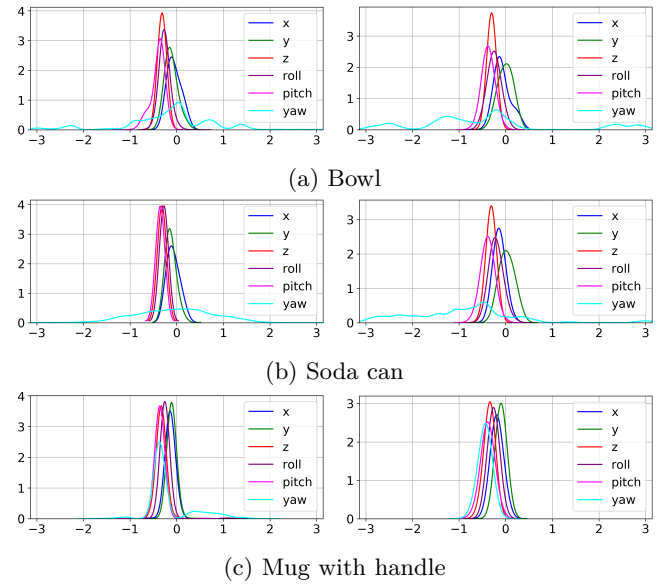
(b) Soda can

(c) Mug with handle

Fig. 3: Visualization of the pose distribution estimated by a kernel density estimator (KDE) for the baseline (left) and Bayesian ICP (right). Both bowl (a) and soda (b) can are rotationally symmetric around their yaw axis which is captured by the broad yaw distribution. This is in contrast to the mug with handle (c) which has no symmetries, resulting in all distributions being peaked.

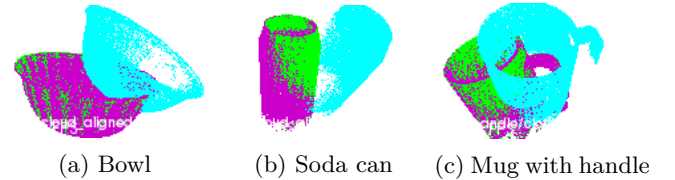(a) Bowl   (b) Soda can   (c) Mug with handle

Fig. 4: Point clouds of objects used in the experiments, (cyan) source, (magenta) reference, and (green) solution. Both (a) and (b) are under-constrained as they are symmetric around their yaw axis, while (c) has no symmetries.

and standard deviation of the KL divergence between the baseline distribution and the distribution estimated by the two methods, with lower values indicating a closer match with the baseline distribution. The first four objects are symmetric around their yaw axis while the last two have no symmetries. We can see that overall our proposed method outperforms Online-ICP when it comes to matching the distribution. Especially interesting are the yaw values, where both methods tend to have higher values, however, the gap between Online-ICP and our proposed method widens significantly on objects with symmetries. This is explained by the fact that our method captures the uncertainty in the distribution as shown in previous sections, while Online-ICP tends to always predict an extremely peaked covariance matrix with values on the order of 10^{-7}, even for yaw on objects with symmetries. This overconfident covariance estimate is also the reason that allows Online-ICP to obtain KL divergences of 0 in cases where the baseline distribution is very peaked.

TABLE I: Quality Comparison of the Bayesian ICP against the Online-ICP using KL Divergence

| θ	Method	Objects					
		Bowl	Soda Can	Mug w/o handle	Table top	Mug w/ handle	Complete table
x	Ours	0.387 ± 0.199	0.118 ± 0.078	0.189 ± 0.051	0.619 ± 0.044	0.133 ± 0.049	0.086 ± 0.011
	[7]	0.731 ± 0.000	0.078 ± 0.000	0.104 ± 0.000	1.938 ± 0.000	0.100 ± 0.000	0.643 ± 0.000
y	Ours	0.373 ± 0.311	0.163 ± 0.168	0.218 ± 0.112	1.117 ± 0.222	0.092 ± 0.019	0.082 ± 0.016
	[7]	0.254 ± 0.000	0.312 ± 0.000	1.596 ± 0.000	1.845 ± 0.000	0.077 ± 0.000	0.062 ± 0.000
z	Ours	0.007 ± 0.003	0.020 ± 0.006	0.015 ± 0.004	0.015 ± 0.000	0.023 ± 0.011	0.008 ± 0.001
	[7]	0.000 ± 0.000	0.000 ± 0.000	0.055 ± 0.000	0.027 ± 0.000	0.018 ± 0.000	0.000 ± 0.000
roll	Ours	0.040 ± 0.017	0.063 ± 0.055	0.105 ± 0.019	5.947 ± 0.828	0.425 ± 0.022	0.012 ± 0.003
	[7]	0.078 ± 0.000	0.000 ± 0.000	0.166 ± 0.000	7.889 ± 0.000	0.571 ± 0.000	0.000 ± 0.000
pitch	Ours	0.084 ± 0.038	0.051 ± 0.045	0.335 ± 0.016	1.099 ± 0.023	0.098 ± 0.027	0.015 ± 0.004
	[7]	0.454 ± 0.000	0.000 ± 0.000	0.572 ± 0.000	1.568 ± 0.000	0.081 ± 0.000	0.002 ± 0.000
yaw	Ours	8.849 ± 6.113	6.620 ± 3.251	5.467 ± 4.926	0.842 ± 0.254	6.457 ± 1.062	1.148 ± 0.283
	[7]	39.689 ± 0.001	32.955 ± 0.001	370.697 ± 0.004	21.917 ± 0.000	13.438 ± 0.001	38.848 ± 0.000
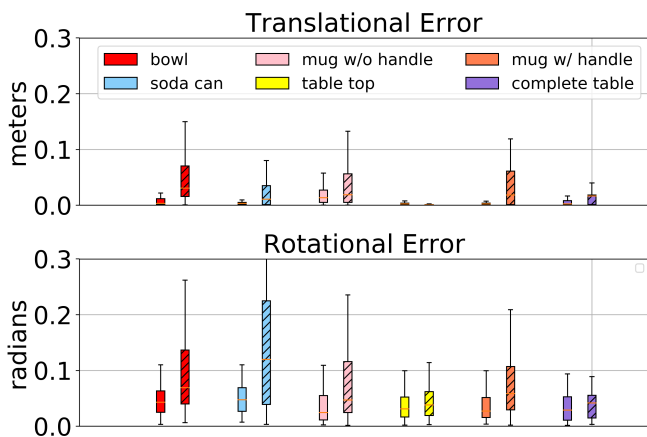
Fig. 5: Side-by-side comparison of the error in translation (top) and rotation (bottom) of Bayesian ICP (left bars) and SGD-ICP (right hatched bars) for random initial true transforms over 50 runs. Covering the solution space with samples allows Bayesian ICP to slightly improve on SGD-ICP results.

D. Run-time

A common concern with MCMC based methods is their runtime and as such we provide an evaluation on the runtime of our proposed method. The time needed to obtain 1000 samples is 8.88 ± 3.59 s, where the variability comes from the size of the point clouds and number of potentially matching points in each mini-batch. However, the sampling based nature and i.i.d. assumption of MCMC allow our method to be trivially parallelized. A straight forward OpenMP based implementation improves the runtime to 6.392 ± 0.731 s using two cores and 3.298 ± 0.573 s using four cores. Synchronization overhead prevents perfect linear scaling between one and two cores, however, the speed gains are significant. This shows that the proposed method can produce high quality pose distribution estimates in a few seconds. For comparison, obtaining 1000 samples for a baseline distribution using standard ICP methods takes several hours.

E. Pose estimation accuracy

Finally, we investigate the quality of the final pose estimate of the proposed method and provide comparison with SGD-ICP solutions. For each object both methods start with a random initial transformation drawn from $[-1, 1]$ m and $[-0.3, 0.3]$ rad for the translation and rotation component respectively. This is repeated 50 times and the error with respect to the ground truth transformation are shown in Fig. 5, Bayesian ICP (left bars) and SGD-ICP (right hatched bars). Both methods obtain good results though in some situations Bayesian ICP obtains higher quality solutions. An explanation for this is that the sampling nature of the method forces exploration of the solution space, thus avoiding initial local minima which are selected by SGD-ICP.

VI. CONCLUSION

This paper introduced a novel ICP method, called Bayesian ICP, which computes not only the expected transformation between two point clouds but also estimates the full pose distribution. This is achieved by leveraging the stochastic nature of SGD-ICP which is combined with stochastic gradient Langevin dynamics, an efficient MCMC method. The resulting algorithm is capable of producing high-quality posterior distributions in a few seconds compared to hours, that a standard MCMC approach would require. Extensive experiments evaluate the impact of parameter choices and showcase the ability of our method to produce accurate posterior estimates as well as expectations of the transformation. The ability to provide high quality pose distributions has many applications in robotics from localization to registration. In future work, we plan to apply Bayesian ICP to 3D SLAM problems, integrating non-parametric filtering techniques such as particle filters into our framework. Another potential application is grasping where Bayesian ICP can be used to estimate the position of objects to be grasped, and the uncertainty estimate can be passed to a motion planner to place the gripper in a more likely position for a successful grasp.

References

[1] F. Afzal Maken, F. Ramos, and L. Ott. Speeding up iterative closest point using stochastic gradient descent. In *IEEE International Conference on Robotics and Automation*, 2019.

[2] O. Bengtsson and A. J. Baerveldt. Robot localization based on scan-matching - estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems*, 2003.

[3] P. Besl and N.D. McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, 1992.

[4] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *IROS*, 2003.

[5] S. Bonnabel and F. Barczyk, M.and Goulette. On the covariance of ICP-based scan-matching techniques. *Proceedings of the American Control Conference*, 2016.

[6] M. Bosse and R. Zlot. Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM. *I. J. Robotic Res.*, 2008.

[7] A. Censi. An accurate closed-form estimate of ICP's covariance. *In IEEE International Conference on Robotics and Automation*, 2007.

[8] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *In Proceedings of IEEE International Conference on Robotics and Automation*, 1991.

[9] J. Glover and S. Popovic. Bingham procrustean alignment for object detection in clutter. *IEEE International Conference on Intelligent Robots and Systems*, 2013.

[10] T. M. Iversen, A. G. Buch, and D. Kraft. Prediction of icp pose uncertainties using monte carlo simulation with synthetic depth images. *IEEE International Conference on Intelligent Robots and Systems*, 2017.

[11] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[12] K. Lai, L. Bo, and D. Fox. Unsupervised feature learning for 3d scene labeling. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014.

[13] C. Li, C. Chen, D. Carlson, and L. Carin. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

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

[15] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2010.

[16] F. Pomerleau, F. Colas, and R. Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends in Robotics*, 2015.

[17] S. M. Prakhya, L. Bingbing, Y. Rui, and W. Lin. Covariance of ICP Algorithm A Closed-form Estimate of 3D ICP Covariance. In *Preceedings of 14th IAPR International Conference on Machine Vision Applications (MVA)*, 2015.

[18] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2005.

[19] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, 2001.

[20] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Robotics: Science and Systems*, 2009.

[21] Y. W. Teh, A. H. Thiery, and S. J. Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *J. Mach. Learn. Res.*, 2016.

[22] T. Tieleman and G. Hinton. Coursera: Neural netwroks for machine learning (lecture 6.5 - rmsprop), 2012.

[23] R. VON MISES. Uber die "ganzzahligkeit" der atomgewicht und verwandte fragen. *Physikalische Z*, 1918.

[24] M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011.

[25] G. Barrie Wetherill. *Regression analysis with applications*. London: Chapman and Hall, cop., 1986.