Probabilistic pose estimation using a Bingham distribution-based linear filter

Rangaprasad Arun Srivatsan (iD)**, Mengyun Xu, Nicolas Zevallos and Howie Choset**

Abstract

Pose estimation is central to several robotics applications such as registration, hand–eye calibration, and simultaneous localization and mapping (SLAM). Online pose estimation methods typically use Gaussian distributions to describe the uncertainty in the pose parameters. Such a description can be inadequate when using parameters such as unit quaternions that are not unimodally distributed. A Bingham distribution can effectively model the uncertainty in unit quaternions, as it has antipodal symmetry, and is defined on a unit hypersphere. A combination of Gaussian and Bingham distributions is used to develop a truly linear filter that accurately estimates the distribution of the pose parameters. The linear filter, however, comes at the cost of state-dependent measurement uncertainty. Using results from stochastic theory, we show that the state-dependent measurement uncertainty can be evaluated exactly. To show the broad applicability of this approach, we derive linear measurement models for applications that use position, surface-normal, and pose measurements. Experiments assert that this approach is robust to initial estimation errors as well as sensor noise. Compared with state-of-the-art methods, our approach takes fewer iterations to converge onto the correct pose estimate. The efficacy of the formulation is illustrated with a number of examples on standard datasets as well as real-world experiments.

Keywords

Kalman filter, pose estimation, Bingham distribution, registration, Bayes rule

1. Introduction

Several applications in robotics require the estimation of pose (translation and orientation) between a model frame and a sensor frame. Examples include medical image registration (Moghari and Abolmaesumi, 2007), manipulation (Engelhard et al., 2011), hand–eye calibration (Faion et al., 2012), and navigation (Engelhard et al., 2011). Filtering-based online pose estimation techniques have particularly been a popular choice due to their ability to adapt to noisy sensor measurements. Most of the prior work on online pose estimation linearize the nonlinear measurement model. This results in inaccurate estimates especially when the initial pose estimate is erroneous. To address this issue, recently Srivatsan et al. (2016) used dual quaternions and developed a linear Kalman filter that is robust to initial pose errors. In this work, a Bingham distribution is used to model the uncertainty in the rotation parameters, and a linear measurement model is adapted from Srivatsan et al. (2016) to develop a linear filter for online pose estimation.

Gaussian distributions can be used to model uncertainty, but are largely limited to linear spaces and states that are unimodal in distribution. Although much work has been done to counteract this assumption, the bottom line is that linearizations are still approximations. We believe that to

obtain optimal state estimates, it is critical to model uncertainties using the appropriate distributions that exploit the fundamental structure of the parameter space. Our prior work (Srivatsan et al., 2016), as well as the work of others (LaViola Jr, 2003; Marins et al., 2001), used unit quaternions as the underlying space and Gaussians to model the uncertainty in their distribution. Gaussian distributions do not consider the structure of the underlying space, i.e., antipodal symmetry introduced by $\widetilde{q} = -\widetilde{q}$ (Kurz et al., 2013). This work introduces an online pose estimation method that uses a Bingham distribution and a Gaussian distribution to robustly and accurately estimate the rotation and translation respectively.

The Bingham distribution is defined on a unit hypersphere and captures the antipodal symmetry of the distribution of unit quaternions (Bingham, 1974) (see Figure 1). When compared with prior methods, the use of the

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Corresponding author:
Rangaprasad Arun Srivatsan, Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA.
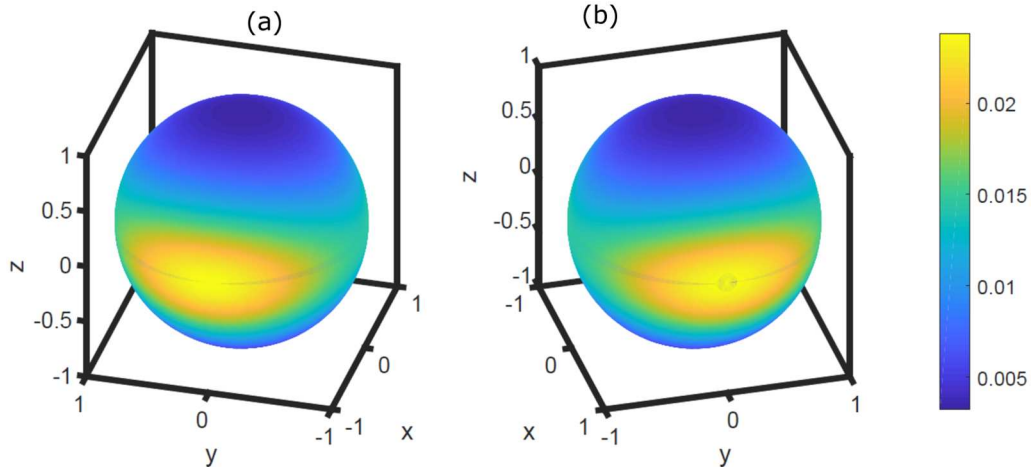Email: rarunsrivatsan@cmu.edu

Fig. 1. A 3D Bingham distribution: $c = \frac{1}{N} \exp(\mathbf{v}^{\mathrm{T}} \mathbf{M} \mathbf{Z} \mathbf{M}^{\mathrm{T}} \mathbf{v})$, where $\mathbf{M} = \mathbf{I}_{3\times3}$, $\mathbf{Z} = diag(0, -0.5, -2)$, and $\mathbf{v} = (x, y, z)^{\mathrm{T}}$, $\mathbf{v}^{\mathrm{T}} \mathbf{v} = 1$. The colors on the sphere show the probability value. (a) Mode at $\mathbf{v} = (1, 0, 0)^{\mathrm{T}}$ and (b) mode at $\mathbf{v} = (-1, 0, 0)^{\mathrm{T}}$. More details can be obtained from Section 3.2.

Bingham distribution results in a formulation that is accurate and has shorter computation time, because there is no normalization step or projection onto a hyper-sphere.

Another advantage of our approach compared with existing methods (Moghari and Abolmaesumi, 2007; Pennec and Thirion, 1997; Srivatsan et al., 2016) is the ability to update the pose not only using point measurements, but also using surface-normal and pose measurements as well as simultaneous multiple measurements (as obtained from a stereo camera or lidar). We extend our previous work on Bingham distribution-based filter (Srivatsan et al., 2017) by developing a filter that uses pose measurements, for applications such as hand–eye calibration (Ackerman et al., 2013; Faion et al., 2012).

Inspired by Horn (1987), this work estimates pose by decoupling orientation from translation estimation. The method uses a Bingham distribution-based filtering (BF) for orientation estimation and a Kalman filter for translation estimation. Although there has been some recent work on using the BF for orientation estimation (Gilitschenski et al., 2016; Kurz et al., 2013), there are some key differences compared with our approach. First, prior work assumes that the state and measurements both are unit quaternions. Second, prior works deal with nonlinear measurement models (Lefebvre et al., 2002), hence requiring approximations introduced by linearization or deterministic sampling such as Steinbring and Hanebeck (2013), Gilitschenski et al. (2016). This results in computation of the normalization constant which is known to be expensive (Glover et al., 2012; Kurz et al., 2013). On the other hand, our approach does not limit measurements to be unit quaternions and bypasses the computation of normalization constant by using a linear measurement model. The computation of normalization constant would still be required to find the covariance of the orientation parameters, which is not part of the algorithm itself.

Faugeras and Hebert (1986) and Walker et al. (1991) have also derived linear models for pose estimation. However, they estimate pose in a deterministic setting and do not provide any guidelines for estimating the uncertainties associated with pose. Following Choukroun et al. (2006) and Srivatsan et al. (2016), in this work we use results from stochastic filtering theory to derive exact expressions for the pose uncertainties.

We first derive the theory for applications where the data association between the model and the sensor measurements is known. Following this, we extend the approach to applications with unknown data association. To deal with unknown data association, we use k-d tree search (Besl and McKay, 1992) and principal-direction tree (PD-tree) search (Verma et al., 2009) in our prior work (Srivatsan et al., 2017). As an extension to our prior work (Srivatsan et al., 2017), in this work we introduce a new look-up table-based approach for fast computations when the data association is unknown. We show that the computational time taken by this approach is a couple of orders of magnitude less than tree search. Although our prior work only considered static pose estimation, in this work we take advantage of the reduced computational time to recursively run the filter for dynamic pose estimation.

This paper is an improved and extended version of our prior work (Srivatsan et al., 2017). These improvements include a more detailed discussion on the derivation of the equations for the linear filter including some visualizations to provide intuition in the appendix. We extend our prior formulation to use pose measurements in addition to point and surface normal measurements in Section 4. We present a new approach to deal with unknown data association in Section 4.4. Compared with Srivatsan et al. (2017), we provide comparison with several state-of-the-art methods for simulation and real-world experiments in Section 5. Although this work focuses on static pose estimation,

the ideas presented can be adapted to dynamic pose estimation. We present some preliminary results for dynamic pose estimation in Section 5.2.4.

2. Related work

2.1. Batch processing approaches

Pose estimation has been of interest for a long time in the robotics literature. Much of the early literature deals with collecting all sensor measurements and processing them offline in a batch to estimate the pose. Horn (1987) developed a least-squares implementation for pose estimation with known point correspondence. Besl and McKay (1992) introduced the iterative closest point (ICP), which extends Horn's methods for unknown point correspondence by iteratively estimating point correspondence and performing least-squares optimization. Several variants of the ICP have been developed (Izatt et al., 2017; Rusinkiewicz and Levoy, 2001; Yang et al., 2013; Zhou et al., 2016). Estépar et al. (2004), Segal et al. (2009), and Billings et al. (2015) further generalized the ICP by incorporating measurement noise uncertainties.

Orientation data (surface normals) has been used in addition to point data for registration in prior works. The approach of Pulli (1999) uses surface-normal information to filter out measurements during the correspondence stage. Münch et al. (2010) used point and surface-normal measurements in both the correspondence and minimization step. Billings and Taylor (2014) have recently developed iterative most likely oriented point registration (IMLOP), a probabilistic framework to estimate pose using surface-normal and position measurements, incorporating measurement uncertainty in both the correspondence and minimization step.

In addition to point and surface-normal measurements there have also been several approaches developed for pose estimation using pose measurements for applications such as hand–eye calibration. The works of Tsai and Lenz (1989), Horaud and Dornaika (1995), and Daniilidis (1999) formulate hand–eye calibration as a least-squares estimation problem assuming complete knowledge of measurement correspondence. More recently, Ackerman et al. (2013) estimate the pose with unknown correspondence.

2.2. Probabilistic sequential estimation

Probabilistic sequential estimation approaches provide sequential state updates based on a continuous stream of sensor measurements. The uncertainty in the state variables is often modeled using probability density functions (PDFs) and the parameters of the PDF are updated after each measurement. In contrast to batch estimation methods, where there is no indication of when to stop collecting measurements, convergence of the state estimate and decrease in the state uncertainty provides clear indication of when to stop collecting measurements.

2.2.1. Gaussian filtering approaches. Several sequential estimation methods are based on Kalman filters, which model the states and measurements using Gaussian distributions (Faion et al., 2012; Hauberg et al., 2013; Moghari and Abolmaesumi, 2007; Pennec and Thirion, 1997; Srivatsan et al., 2016). Kalman filters by construction provide optimal state estimates when the process and measurement models are linear and the states and measurements are Gaussian distributed (Kalman, 1960). Pose estimation, however, is inherently a nonlinear problem, and hence linear Kalman filters produce poor estimates (Hauberg et al., 2013; Pennec and Thirion, 1997). Several variants of the Kalman filter have been introduced to handle the nonlinearity. Extended Kalman filter (EKF)-based filters perform first-order linear approximations of the nonlinear models and produce estimates that are known to diverge in the presence of high initial estimation errors (Moghari and Abolmaesumi, 2007). Unscented Kalman filter (UKF)-based methods do not linearize the models but instead utilize evaluation at multiple points, which can be expensive for a high-dimensional system such as $SE(3)$ (Hauberg et al., 2013; Moghari and Abolmaesumi, 2007). In addition, UKF-based methods require tuning a number of parameters, which can be unintuitive.

2.2.2. Non-Gaussian filtering approaches. There has been some recent work in robotics towards the use of alternative distributions to model the noise on rotations for pose estimation problems. For example, Langevin distributions have been used for pose estimation by Carlone and Censi (2014) and Rosen et al. (2016). Gilitschenski et al. (2016) have recently developed a Bingham distribution-based recursive filtering approach for orientation estimation. Glover et al. (2012) used Bingham distribution to describe the orientation features, whereas Gilitschenski et al. (2014) used this distribution for planar pose estimation. Our work takes inspiration from these works for modeling the uncertainty in the orientation using Bingham distribution. In a more recent work, Fan et al. (2017) used a constrained Kalman filter with dual quaternions as state vector. Although the authors do not explicitly mention the use of Bingham distribution, the update equations of the constrained Kalman filter are identical to the use of a Bingham filter as shown by Srivatsan et al. (2017).

The use of Bingham distributions to model uncertainties in rotation parameters is a very valuable tool that has been largely under-utilized by the robotics community, as also noted by Glover et al. (2012). One of the important reasons for this is the difficulty in computing the normalization constant as well as performing expensive convolution operation over the distributions (Bingham, 1974).

2.2.3. Alternate parameterizations for filtering. Prior work also has looked at several parameterizations of $SE(3)$ that would improve the performance of the filters. Hauberg et al.

(2013) confined the state variables over a known Riemannian manifold and use a UKF to estimate the pose. Quaternions are used to parametrize $SO(3)$ and the state is estimated using an EKF in Marins et al. (2001) and UKF in LaViola Jr (2003). An iterated EKF with dual quaternions to parameterize the pose has been used in Goddard and Abidi (1998).

2.2.4. Linear filtering approach. Srivatsan et al. (2016) have recently developed a linear Kalman filter for pose estimation using dual quaternions and pairwise measurement update. Although this method has been shown to be robust to errors in initial state estimate and sensor noise, it has a few drawbacks. (1) The uncertainty in the quaternions used for orientation estimate is modeled using Gaussians that do not consider the condition that $\widetilde{q}$ and $-\widetilde{q}$ represent the same rotation. (2) The filter by itself does not produce unit-quaternion estimates and, hence, after each estimate, a projection step is used to normalize the state. The projection step introduces an additional error that is not accounted for in the uncertainty estimate (Julier and LaViola, 2007). In addition, such a projection would have a large error if the estimated state had a near zero norm. (3) The approach only requires pairs of measurements per update. However, in many practical applications such as image registration, several ($\approx 10^4$) measurements are available for processing in each update step, and a pairwise update could be very inefficient and time consuming.

3. Mathematical background

Before going into the description of the linear filter for pose estimation, we provide a brief introduction to the concepts of quaternions and the Bingham distribution.

3.1. Quaternions

Although there are many representations for $SO(3)$ elements such as Euler angles, Rodrigues parameters, and axis angles, in this work we use unit quaternions. We prefer the quaternions because their elements vary continuously over the unit sphere $\mathcal{S}^3$ as the orientation changes, avoiding discontinuous jumps (inherent to three-dimensional parameterizations).

A quaternion $\widetilde{q}$ is a 4-tuple:

$$\widetilde{q} = (q_0, q_1, q_2, q_3)^{\mathrm{T}}, \quad \widetilde{q} \in \mathbb{R}^4$$

where q_0 is the scalar part and $\mathrm{vec}(\widetilde{q}) = q = (q_1, q_2, q_3)^{\mathrm{T}}$ is the vector part of the quaternion. Sometimes an alternate convention is used where $\widetilde{q} = (q_1, q_2, q_3, q_0) = (\mathrm{vec}(\widetilde{q}), \mathrm{scalar}(\widetilde{q}))$ (Bingham, 1974).

3.1.1. Quaternion multiplication. Multiplication of two quaternions $\widetilde{p}$ and $\widetilde{q}$ is given by

$$\widetilde{p} \odot \widetilde{q} = \underbrace{\begin{bmatrix} p_0 & -p^{\mathrm{T}} \\ p & p^{\times} + p_0 I_3 \end{bmatrix}}_{F_1(\widetilde{p})} q = \underbrace{\begin{bmatrix} q_0 & -q^{\mathrm{T}} \\ q & -q^{\times} + q_0 I_3 \end{bmatrix}}_{F_2(\widetilde{q})} p \quad (1)$$

where $\odot$ is the quaternion multiplication operator and $[v]^{\times}$ is the skew-symmetric matrix formed from the vector v.

3.1.2. Quaternion conjugate. Given a quaternion $\widetilde{q}$, its conjugate $\widetilde{q}^*$ can be written as

$$\widetilde{q}^* = (q_0, -q_1, -q_2, -q_3)^{\mathrm{T}} \quad (2)$$

The norm of a quaternion is

$$\|\widetilde{q}\| = \sqrt{\mathrm{scalar}(\widetilde{q} \odot \widetilde{q}^*)} \quad (3)$$

3.1.3. Unit quaternions. A unit quaternion is one with $\|\widetilde{q}\| = 1$. Unit quaternions can be used to represent rotation about an axis (denoted by the unit vector k) by an angle $\theta \in [-\pi, \pi]$ as follows

$$\widetilde{q} = \left(\cos\left(\frac{\theta}{2}\right), k^{\mathrm{T}} \sin\left(\frac{\theta}{2}\right) \right)^{\mathrm{T}} \quad (4)$$

As rotating about k axis by θ is the same as rotating about $-k$ axis by $-\theta$, $\widetilde{q}$ and $-\widetilde{q}$ both represent the same rotation.

3.1.4. Rotation using quaternions. A point b can be rotated by a quaternion $\widetilde{q}$ to obtain a new point a as shown,

$$\widetilde{a} = \widetilde{q} \odot \widetilde{b} \odot \widetilde{q}^* \quad (5)$$

where $\widetilde{a} = (0, a^{\mathrm{T}})^{\mathrm{T}}$ and $\widetilde{b} = (0, b^{\mathrm{T}})^{\mathrm{T}}$ are quaternion representations of a, b respectively.

3.2. Bingham distribution

The Bingham distribution was introduced by Bingham (1974) as an extension of the Gaussian distribution, conditioned to lie on the surface of a unit hyper-sphere. The Bingham distribution is widely used in analyzing palaeomagnetic data (Kunze and Schaeben, 2004; Onstott, 1980), computer vision (Haines and Wilson, 2008), and directional statistics (Bingham, 1974) and recently in robotics (Gilitschenski et al., 2014, 2016; Glover et al., 2012; Kurz et al., 2013).

Definition 1. *Let $\mathcal{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\| = 1\} \subset \mathbb{R}^d$ be the unit hypersphere in $\mathbb{R}^d$. The probability density function $f : \mathcal{S}^{d-1} \to \mathbb{R}$ of a Bingham distribution is given by*

$$f(x) = \frac{1}{N} \exp(x^{\mathrm{T}} M Z M^{\mathrm{T}} x)$$

where $M \in \mathbb{R}^{d \times d}$ is an orthogonal matrix ($MM^{\mathrm{T}} = M^{\mathrm{T}} M = I^{d \times d}$), $Z = diag(z_1, \ldots, z_d) \in \mathbb{R}^{d \times d}$ with $z_1 \geq z_2 \geq \cdots \geq z_d$, is known as the concentration matrix, and N is a normalization constant.

3.2.1. Mode of the distribution. It can be shown that adding a multiple of the identity matrix $\boldsymbol{I}^{d \times d}$ to $\boldsymbol{Z}$ does not change the distribution (Bingham, 1974):

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= \frac{1}{N_1} \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M}(\boldsymbol{Z} + \lambda \boldsymbol{I}) \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \\
&= \frac{1}{N_1} \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M}(\lambda \boldsymbol{I}) \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \\
&= \frac{1}{N_1} \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \exp(\lambda \boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \\
&= \frac{1}{N} \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) = f(\boldsymbol{x})
\end{aligned}
$$

where $N = \frac{N_1}{\exp(\lambda)}$, $\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} = 1$, and $\boldsymbol{M} \boldsymbol{M}^{\mathrm{T}} = \boldsymbol{I}$.

Thus, we conveniently force the first entry of $\boldsymbol{Z}$ to be zero (Bingham, 1974). Because it is possible to swap columns of $\boldsymbol{M}$ and the corresponding diagonal entries in $\boldsymbol{Z}$ without changing the distribution, we can enforce $z_1 \geq \cdots \geq z_d$, where $z_1 = 0$. This representation allows us to obtain the mode of the distribution very easily by taking the first column of $\boldsymbol{M}$. The proof for this is shown below. The mode of $f(\boldsymbol{x})$ is given by

$$
\begin{aligned}
\boldsymbol{x}^* &= \operatorname*{argmax}_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \text{subject to } |\boldsymbol{x}| = 1 \\
&= \operatorname*{argmax}_{\boldsymbol{x}} \boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x} - \lambda(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} - 1)
\end{aligned}
$$

where λ is the Lagrangian multiplier. Here $\boldsymbol{x}^*$ is obtained by solving for $\boldsymbol{x}$ from the following,

$$
\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{x}} \left(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x} - \lambda(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{x} - 1) \right) = 0 \\
&\Rightarrow \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x} - \lambda \boldsymbol{x} = 0
\end{aligned}
$$

Thus, the Lagrangian multiplier λ is the largest eigenvalue and $\boldsymbol{x}^*$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}}$. From the construction of $\boldsymbol{Z}$, the largest eigenvalue is zero (which appears in the first column of $\boldsymbol{Z}$) and the corresponding eigenvector is the first column of $\boldsymbol{M}$.

Note that sometimes an alternate convention is used in literature, wherein $\boldsymbol{Z}$ is chosen such that the last entry of $\boldsymbol{Z}$ is zero and the last column of $\boldsymbol{M}$ is chosen as the mode of the distribution (Kurz et al., 2013; Bingham, 1974).

3.2.2. Normalization constant. The normalization constant N is given by

$$
\begin{aligned}
N &= \int_{\mathcal{S}^{d-1}} \exp(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}) \, d\boldsymbol{x} \\
&= \int_{\mathcal{S}^{d-1}} \exp(\boldsymbol{s}^{\mathrm{T}} \boldsymbol{Z} \boldsymbol{s}) \, d\boldsymbol{s}
\end{aligned}
$$

The matrix $\boldsymbol{M}$ does not affect the normalization constant (Bingham, 1974). An intuition for the same is that $\boldsymbol{Z}$ is an indicator of the spread of the distribution around the modes and $\boldsymbol{M}$ is an indicator of the location of the modes and the direction of the dispersion with respect to the modes.
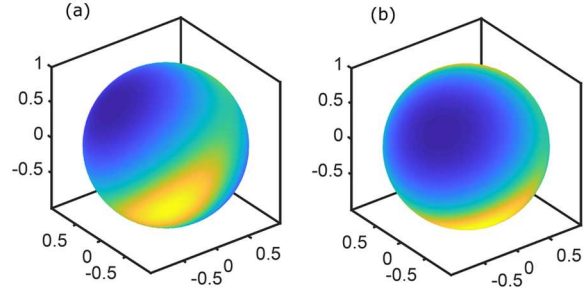

Fig. 2. Two Bingham distributions with $\boldsymbol{Z} = diag(0, -0.5, -2)$, (a) $\boldsymbol{M} = \begin{pmatrix} 0.5322 & -0.4953 & -0.6866 \\ 0.7747 & 0.6120 & 0.1591 \\ 0.3415 & -0.6166 & 0.7094 \end{pmatrix}$, (b) $\boldsymbol{M} = \begin{pmatrix} 0.2603 & -0.6941 & -0.6712 \\ 0.6578 & 0.6364 & -0.4029 \\ 0.7068 & -0.3366 & 0.6222 \end{pmatrix}$

Hence, for a given $\boldsymbol{Z}$, changing $\boldsymbol{M}$ results in an identical distribution with a different location of the modes and orientation of the dispersion, which does not affect the integration. Figure 2 shows two choices of $\boldsymbol{M}$ with the same $\boldsymbol{Z}$ that results in identical Bingham distributions that appear to be rotated with respect to each other.

Computation of the normalization constant is difficult and often one resorts to some form of approximation such as saddle point approximations or precomputed lookup tables (see Glover et al., 2012 and the references therein).

3.2.3. Antipodal symmetry. An example of the PDF for two dimensions ($d = 3$) is shown in Figure 1. The PDF is antipodally symmetric, i.e., $f(\boldsymbol{x}) = f(-\boldsymbol{x})$ holds for all $\boldsymbol{x} \in \mathcal{S}^{d-1}$. The antipodal symmetry is important when dealing with distribution of unit quaternions, because the $\widetilde{\boldsymbol{q}}$ and $-\widetilde{\boldsymbol{q}}$ describe the same rotation. The Bingham distribution with $d = 4$ is used to describe the uncertainty in the space of the unit quaternions.

3.2.4. Product of two Bingham densities. Similar to a Gaussian, the product of two Bingham PDFs is a Bingham distribution, which can be rescaled to form a PDF (Kurz et al., 2013). Consider two Bingham distributions $f_i(\boldsymbol{x}) = \frac{1}{N_i} \exp\left(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M}_i \boldsymbol{Z}_i \boldsymbol{M}_i^{\mathrm{T}} \boldsymbol{x}\right)$, $i = 1, 2$. Then,

$$
\begin{aligned}
f_1(\boldsymbol{x}) &\cdot f_2(\boldsymbol{x}) \\
&= \frac{1}{N_1 N_2} \exp(\boldsymbol{x}^{\mathrm{T}} \underbrace{\left(\boldsymbol{M}_1 \boldsymbol{Z}_1 \boldsymbol{M}_1^{\mathrm{T}} + \boldsymbol{M}_2 \boldsymbol{Z}_2 \boldsymbol{M}_2^{\mathrm{T}}\right)}_{A} \boldsymbol{x}) \\
&\propto \frac{1}{N} \exp\left(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^{\mathrm{T}} \boldsymbol{x}\right) \quad (6)
\end{aligned}
$$

where N is the new normalization constant after renormalization, $\boldsymbol{M}$ is composed of the unit eigenvectors of A. $\boldsymbol{Z} = \boldsymbol{D} - \boldsymbol{D}_{11} \boldsymbol{I}_{d \times d}$ where $\boldsymbol{D}$ has the eigenvalues of A (sorted in descending order) and $\boldsymbol{D}_{11}$ refers to the largest eigenvalue.

3.2.5. Calculating the covariance. Even though a Bingham distributed random vector $\boldsymbol{x}$ only takes values on the

unit hyper-sphere, it is still possible to compute a covariance matrix in $\mathbb{R}^d$ (Kurz et al., 2013), which is given by

$$Cov(\boldsymbol{x}) = E(\boldsymbol{x}^2) - E(\boldsymbol{x})^2 \tag{7}$$

$$= \boldsymbol{M}\left(diag\left(\frac{\frac{\partial N(\boldsymbol{Z})}{\partial z_1}}{N(\boldsymbol{Z})}, \cdots, \frac{\frac{\partial N(\boldsymbol{Z})}{\partial z_d}}{N(\boldsymbol{Z})}\right)\right)\boldsymbol{M}^{\mathrm{T}} \tag{8}$$

This is equivalently the covariance of a normally distributed $\boldsymbol{x}$ sampled from $\mathcal{N}\left(0, -0.5\left(\boldsymbol{M}(\boldsymbol{Z}+\lambda\boldsymbol{I})\boldsymbol{M}^{\mathrm{T}}\right)^{-1}\right)$, given $|\boldsymbol{x}| = 1$, where $\lambda \in \mathbb{R}$ can be arbitrarily chosen as long as $(\boldsymbol{Z}+\lambda\boldsymbol{I})$ is negative definite (Gilitschenski et al., 2016). Figure 3 shows two choices of λ resulting in the same Bingham distribution. Without loss of generality $\lambda = min(z_i)$ is chosen in this work.

3.2.6. Composition of two Bingham distributions. Composition can be useful when we want to disturb a system, whose uncertainties are modeled with a Bingham distribution, with a Bingham distributed noise. Unfortunately, the Bingham distribution is not closed under composition and we can only approximate the composition as a Bingham (Bingham, 1974). Although the authors are not aware of a general approach for composing n-dimensional Bingham distributions, for the case of S^1 and S^3, prior work provides a method for finding the parameters of the approximate Bingham distribution obtained by composition (Glover and Kaelbling, 2013; Kurz et al., 2013).

4. Problem Formulation

In this work we consider pose estimation applications that use– 1) position measurements, 2) position and surface-normal measurements, and 3) pose measurements. We derive linear models for all these cases in this section assuming the data association between the model and the sensor measurements is known. Following that we shall discuss the extension of our approach to applications where the data association is unknown.

4.1. Position Measurements

Let $\boldsymbol{a}_i$, $\boldsymbol{b}_i \in \mathbb{R}^3$, $(i = 1, \ldots, n)$ be the locations of n points in two different reference frames whose relative pose is to be estimated. The relation between points $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, is given by

$$\boldsymbol{a}_i = Rot(\widetilde{\boldsymbol{q}})\,\boldsymbol{b}_i + \boldsymbol{t}, \quad i = 1, \ldots, n, \tag{9}$$

where $Rot(\widetilde{\boldsymbol{q}}) \in SO(3)$ is the rotation matrix obtained from the unit quaternion $\widetilde{\boldsymbol{q}} \in \mathbb{R}^4$ and $\boldsymbol{t} \in \mathbb{R}^3$. In an application such as point-registration, $\boldsymbol{a}_i$ are points in CAD-model frame and $\boldsymbol{b}_i$ are points in sensor frame respectively.

4.1.1. Update Model. First consider the scenario where points in the sensor frame are obtained one at a time in a sequential manner, as typically observed in the case of
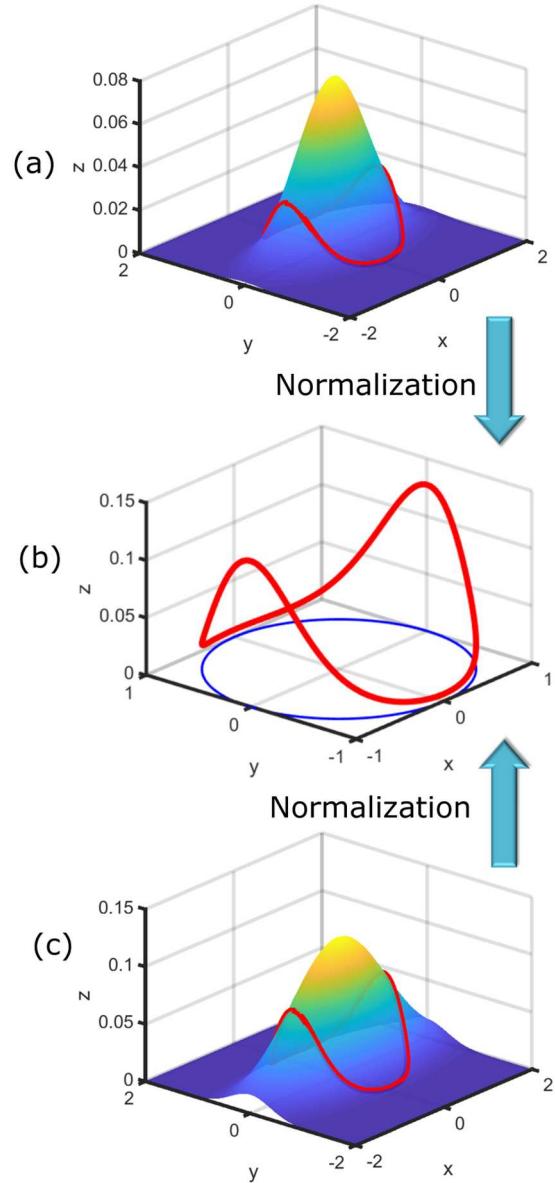

Fig. 3. (a) A Gaussian distribution: $f_1(\boldsymbol{v}) = \frac{1}{N_1}\exp(\boldsymbol{v}^{\mathrm{T}}(diag(0, -3) - 1 \times \boldsymbol{I}_{2\times2})\boldsymbol{v})$. A slice of the Gaussian as made by points lying on a unit circle is shown in red. (b) A Bingham distribution: $f_2(\boldsymbol{v}) = \frac{1}{N_2}\exp(\boldsymbol{v}^{\mathrm{T}}diag(0, -3)\boldsymbol{v})$. (c) A Gaussian distribution: $f_3(\boldsymbol{v}) = \frac{1}{N_3}\exp(\boldsymbol{v}^{\mathrm{T}}(diag(0, -3) - 0.5 \times \boldsymbol{I}_{2\times2})\boldsymbol{v})$. A slice of the Gaussian as made by points lying on a unit circle is shown in red. The slice obtained from the Gaussian distributions in (a) and (c), when normalized result in the same Bingham distribution as shown in (b).

robotic probing Srivatsan et al., (2016a). Similar to (Srivatsan et al., 2016), the equations for updating the pose estimate given a pair of measurements ($n = 2$), are derived. From Eq. 5, Eq. 9 can be rewritten as

$$\widetilde{\boldsymbol{a}}_1 = \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_1 \odot \widetilde{\boldsymbol{q}}^* + \widetilde{\boldsymbol{t}}, \tag{10}$$

$$\widetilde{\boldsymbol{a}}_2 = \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_2 \odot \widetilde{\boldsymbol{q}}^* + \widetilde{\boldsymbol{t}}, \tag{11}$$
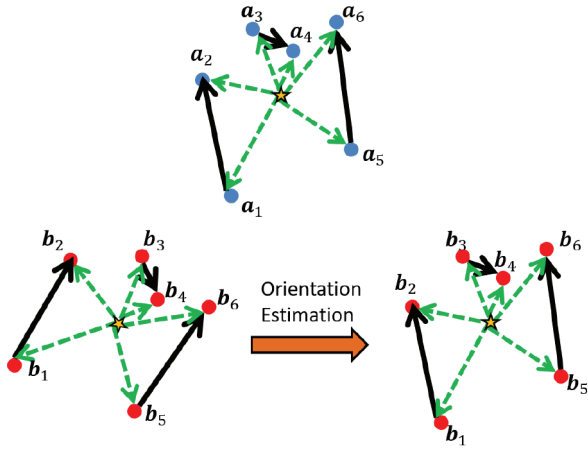
Fig. 4. Blue points (left) indicate a_i and red points (right) indicate b_i. Our approach constructs vectors $a_v^{ij} = (a_i - a_j)$ and $b_v^{ij} = (b_i - b_j)$ as shown by black arrows. The Bingham filter estimates the orientation between the black vectors. A standard implementation of Horn (1987) on the other hand, finds the orientation between the green-dashed vectors. While the green-dashed vectors can only be constructed using a batch of measurements, the black vectors can be constructed from sequential measurements. The black vectors can be considered to be a special case of green-dashed vectors where one pair of data points are considered at a time with one of the points serving as the centroid.

where $\widetilde{q}$ is as defined in Eq. 4 and $\widetilde{t} = (0, t^T)^T$. Subtracting Eq. 11 from Eq. 10,

$$\widetilde{a}_1 - \widetilde{a}_2 = \widetilde{q} \odot (\widetilde{b}_1 - \widetilde{b}_2) \odot \widetilde{q}^*,$$
$$\Rightarrow (\widetilde{a}_1 - \widetilde{a}_2) \odot \widetilde{q} = \widetilde{q} \odot (\widetilde{b}_1 - \widetilde{b}_2), \quad (12)$$

since $\widetilde{q}$ is a unit-quaternion. Using matrix form of quaternion multiplication shown in Eq. 1, Eq. 12 can be rewritten as

$$F_1(\widetilde{a}_1 - \widetilde{a}_2) \widetilde{q} - F_2(\widetilde{b}_1 - \widetilde{b}_2) \widetilde{q} = 0,$$
$$\Rightarrow H(a_1, a_2, b_1, b_2) \widetilde{q} = 0, \quad \text{where} \quad (13)$$

$$H = \begin{bmatrix} 0 & -(a_v - b_v)^T \\ (a_v - b_v) & (a_v + b_v)^\times \end{bmatrix} \in \mathbb{R}^{4\times4}, \quad (14)$$

$a_v = a_1 - a_2$ and $b_v = b_1 - b_2$. Notice that Eq. 13 is a linear equation in terms of $\widetilde{q}$ and is independent of t.

Adding Eq. 10 and Eq. 11,

$$\widetilde{a}_1 + \widetilde{a}_2 = \widetilde{q} \odot (\widetilde{b}_1 + \widetilde{b}_2) \odot \widetilde{q}^* + 2\widetilde{t},$$
$$\Rightarrow \widetilde{t} = \frac{(\widetilde{a}_1 + \widetilde{a}_2) - \widetilde{q} \odot (\widetilde{b}_1 - \widetilde{b}_2) \odot \widetilde{q}^*}{2}. \quad (15)$$

Eq. 13 and Eq. 15 were derived in Srivatsan et al. (2016a) using dual quaternions, however, no geometrical intuition was provided. Fig. 4 provides the geometrical intuition behind the decoupled estimation of $\widetilde{q}$ and t. Estimating the pose between a_i and b_i can be reduced to first estimating the orientation of vectors a_v^{ij} and b_v^{ij} and then estimating

the translation between the centroids of the points. A similar idea is commonly used by Horn (1987). A key difference is that instead of forming vectors $a_v = a_1 - a_2$ and $b_v = b_1 - b_2$, Horn's method uses $a_v = a_1 - a_c$ and $b_v = b_1 - b_c$, where a_c and b_c are the centroids of a_i and b_i respectively.

Further, Eq. 13 is similar to the one used by Faugeras and Hebert (1986). However, in Faugeras and Hebert (1986) uncertainties in sensor measurements were not considered while estimating $\widetilde{q}$. In this work, we model the uncertainty in the sensor measurements a_i, b_i using Gaussian distribution. Let $a_i^s = a_i + \delta a_i$ and $b_i^s = b_i + \delta b_i$, where $(\cdot)^s$ is a sensor measurement, and $\delta(\cdot)$ is the noise as sampled from a zero mean Gaussian, $\mathcal{N}(0, \Sigma^{(\cdot)})$. Eq. 13 can be rewritten as

$$H(a_1, a_2, b_1, b_2) \widetilde{q} = 0,$$
$$H(a_1^s, a_2^s, b_1^s, b_2^s) \widetilde{q} + G(\widetilde{q}) \mu = 0, \quad (16)$$

where $\mu = (\delta a_1, \delta a_2, \delta b_1, \delta b_2)^T$ and

$$G = \begin{bmatrix} -W_1 & W_1 & W_2 & -W_2 \end{bmatrix}, \quad (17)$$

where $W_1, W_2 \in \mathbb{R}^{4\times3}$ are comprised of the last three columns of $F_2(\widetilde{q})$ and $F_1(\widetilde{q})$ respectively.

It can be shown that $G(\widetilde{q}) \mu$ is a zero mean Gaussian noise, $\mathcal{N}(0, Q)$, where the uncertainty Q is obtained analytically. To evaluate Q, we make use of an important result from stochastic filtering theory which is described in Proposition 1 (pp. 90–91 of Jazwinski (2007) and Appendix A of Choukroun et al. (2006)). For the sake of completeness, we prove the proposition in the Appendix A. We also provide a couple of examples that illustrate this proposition.

It is to be noted that Proposition 1 uses uncertainty in $\widetilde{q}$ to evaluate Q. If no prior information about the uncertainty of $\widetilde{q}$ is available, then

$$Q = G(\widetilde{q}) \Sigma^\mu G(\widetilde{q})^T. \quad (18)$$

4.1.2. Linear Filter. In order to obtain an estimate of $\widetilde{q}$ from Eq. 13, we use a Bingham distribution to model the uncertainty in $\widetilde{q}$,

$$p(\widetilde{q}) = \frac{1}{N_1} \exp(\widetilde{q}^T \underbrace{M_{k-1} Z_{k-1} M_{k-1}^T}_{D_1} \widetilde{q}). \quad (19)$$

If the pose was changing with time, then a suitable process model can be employed as shown in Gilitschenski et al. (2016). For now we restrict the analysis to static pose estimation and hence do not consider a process model to evolve the pose estimate over time. Position measurements are obtained, which are in turn used to update the pose estimate. The pose is updated once for every pair of measurements received. The following is the probability of obtaining a sensor measurement z_k, given the state $\widetilde{q}_k$,

$$p(z_k | \widetilde{q}_k) = \frac{1}{N_2} \exp\left(-\frac{1}{2}(z_k - h(\widetilde{q}_k))^T Q_k^{-1}(z_k - h(\widetilde{q}_k))\right), \quad (20)$$

where $\boldsymbol{h}(\widetilde{\boldsymbol{q}}_k)$ is the expected sensor measurement and $\boldsymbol{Q}_k$ is the measurement uncertainty. $\boldsymbol{Q}_k$ is obtained as shown earlier from Proposition 1. Since $\boldsymbol{Q}_k$ is dependent on the state $\widetilde{\boldsymbol{q}}$, we use the current best estimate $\widetilde{\boldsymbol{q}}_{k-1}$ to estimate the uncertainty $\boldsymbol{Q}_k$.

From Eq. 16, we set the measurement to $z_k = 0$ and measurement model to $\boldsymbol{h}(\widetilde{\boldsymbol{q}}_k) = \boldsymbol{H}\widetilde{\boldsymbol{q}}_k$. In a Bayesian update, the state $\widetilde{\boldsymbol{q}}_k$ is updated such that the predicted measurement $\boldsymbol{H}\widetilde{\boldsymbol{q}}_k$ is close to the actual measurement z_k. Setting $z_k = 0$ ensures that the state $\widetilde{\boldsymbol{q}}_k$ is updated so that $\boldsymbol{H}\widetilde{\boldsymbol{q}}_k = z_k = 0$ (as required from Eq. 16).

Since $z_k = 0$ is not a true measurement, it is often referred to as pseudo-measurement in literature (Richards (1995)). For a detailed discussion on deriving update equations using a linear pseudo-measurement model, refer to Chia et al., (1991); Julier and LaViola (2007). Eq. 20 can be rewritten for our case as,

$$p(z_k|\widetilde{\boldsymbol{q}}_k) = \frac{1}{N_2} \exp\left(-\frac{1}{2}(\boldsymbol{H}\widetilde{\boldsymbol{q}}_k)^T \boldsymbol{Q}_k^{-1}(\boldsymbol{H}\widetilde{\boldsymbol{q}}_k)\right),$$
$$= \frac{1}{N_2} \exp\left(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_2 \widetilde{\boldsymbol{q}}_k\right),$$

where $\boldsymbol{D}_2 = \frac{1}{2}\left(-\boldsymbol{H}^T\boldsymbol{Q}_k^{-1}\boldsymbol{H}\right)$. Since $\boldsymbol{Q}_k$ is a positive definite matrix (as required by a Gaussian), $\boldsymbol{D}_2$ is a negative definite matrix. Since $\boldsymbol{D}_2$ is negative definite and $|\widetilde{\boldsymbol{q}}_k^T| = 1$, $p(z_k|\widetilde{\boldsymbol{q}}_k)$ is an unnormalized Bingham distribution in $\widetilde{\boldsymbol{q}}_k$. Note that we assumed the PDF $p(z_k|\widetilde{\boldsymbol{q}}_k)$ to be a Gaussian distribution in z_k, and the algebraic simplification results in the PDF being a Bingham distribution in $\widetilde{\boldsymbol{q}}_k$. Also note that we can use eigen decomposition to obtain parameters of the Bingham distribution $\boldsymbol{D}_2 = \boldsymbol{M}_2\boldsymbol{Z}_2\boldsymbol{M}_2^T$. The parameters $\boldsymbol{M}_2$ and $\boldsymbol{Z}_2$ are not to be confused as being equal to $\boldsymbol{H}$ and $\boldsymbol{Q}_k$, due to the similarity in the form of the equations.

Assuming the measurements are all independent of each other, the updated state given the current state estimate and measurement can be obtained by applying Bayes rule

$$p(\widetilde{\boldsymbol{q}}_k|z_k) \propto p(\widetilde{\boldsymbol{q}}_k)p(z_k|\widetilde{\boldsymbol{q}}_k)$$
$$\propto \frac{1}{N_1} \exp\left(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_1 \widetilde{\boldsymbol{q}}_k\right) \frac{1}{N_2} \exp\left(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_2 \widetilde{\boldsymbol{q}}_k\right) \quad (21)$$
$$\propto \exp\left(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{M}_k \boldsymbol{Z}_k \boldsymbol{M}_k^T \widetilde{\boldsymbol{q}}_k\right). \quad (22)$$

And thus it can be seen that the maximum a posteriori estimate, $p(\widetilde{\boldsymbol{q}}_k|z_k)$, is a Bingham distribution, where $\boldsymbol{M}_k \boldsymbol{Z}_k \boldsymbol{M}_k^T$ is obtained from the product of Binghams as shown in Eq. 6. As mentioned in Sec. 3.2.4, the mode of the distribution $\widetilde{\boldsymbol{q}}_k$, is the first column of $\boldsymbol{M}_k$.

It is worth noting that when no prior uncertainty information is available, $\boldsymbol{Q}_k$ can be obtained from Eq. 18. However, it can be shown that the rank of $\boldsymbol{Q}_k$ (obtained from Eq. 18), is at most three. But Eq. 20 requires $\boldsymbol{Q}_k$ to be invertible. Thus we assume prior uncertainty of $\widetilde{\boldsymbol{q}}$ to be very large in such cases which would allow us to use Proposition 1 and obtain an invertible $\boldsymbol{Q}$. The first few state updates would not produce meaningful results due to high uncertainty in

$\widetilde{\boldsymbol{q}}$. But each update decreases the uncertainty and after a few updates, the filter starts producing meaningful results.

After updating $\widetilde{\boldsymbol{q}}_k$, we estimate $\boldsymbol{t}_k$ from Eq. 15. The prior and likelihood of $\boldsymbol{t}$ are

$$p(\boldsymbol{t}) = \frac{1}{N_3} \exp\left(-\frac{1}{2}(\boldsymbol{t} - \boldsymbol{t}_{k-1})^T (\boldsymbol{\Sigma}_{k-1}^t)^{-1}(\boldsymbol{t} - \boldsymbol{t}_{k-1})\right)$$
$$p(\widetilde{\boldsymbol{q}}_k, \boldsymbol{a}_i, \boldsymbol{b}_i|\boldsymbol{t}) = \frac{1}{N_4} \exp\left(-\frac{1}{2}(\boldsymbol{W}_1\boldsymbol{t} - \boldsymbol{W}_1\boldsymbol{a}_c + \boldsymbol{W}_2\boldsymbol{b}_c)^T \right.$$
$$\left. \boldsymbol{R}_k^{-1}(\boldsymbol{W}_1\boldsymbol{t} - \boldsymbol{W}_1\boldsymbol{a}_c + \boldsymbol{W}_2\boldsymbol{b}_c)\right),$$

where the derivation for the likelihood and the definition of $\boldsymbol{R}_k$, $\boldsymbol{a}_c$, $\boldsymbol{b}_c$ are provided in Appendix B.

We obtain $\boldsymbol{t}_k$ by finding the maximum aposteriori estimate

$$\boldsymbol{t}_k = \underset{\boldsymbol{t}}{\operatorname{argmax}}\, p(\boldsymbol{t})p(\widetilde{\boldsymbol{q}}_k, \boldsymbol{a}_i, \boldsymbol{b}_i|\boldsymbol{t}),$$
$$= \underset{\boldsymbol{t}}{\operatorname{argmin}}(\boldsymbol{t} - \boldsymbol{t}_{k-1})^T (\boldsymbol{\Sigma}_{k-1}^t)^{-1}(\boldsymbol{t} - \boldsymbol{t}_{k-1}) +$$
$$(\boldsymbol{W}_2\boldsymbol{t} - \boldsymbol{W}_2\boldsymbol{a}_c + \boldsymbol{W}_1\boldsymbol{b}_c)^T \boldsymbol{R}_k^{-1}$$
$$(\boldsymbol{W}_2\boldsymbol{t} - \boldsymbol{W}_2\boldsymbol{a}_c + \boldsymbol{W}_1\boldsymbol{b}_c)$$

Upon taking a partial derivative with respect to $\boldsymbol{t}$ and setting it to 0, we get

$$\boldsymbol{t}_k = \left(\left(\boldsymbol{\Sigma}_{k-1}^t\right)^{-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}\boldsymbol{W}_1\right)^{-1}$$
$$\left(\left(\boldsymbol{\Sigma}_{k-1}^t\right)^{-1}\boldsymbol{t}_{k-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}(\boldsymbol{W}_1\boldsymbol{a}_c - \boldsymbol{W}_2\boldsymbol{b}_c)\right) \quad (23)$$

The covariance $\boldsymbol{\Sigma}_k^t$ is obtained from the double derivative

$$\boldsymbol{\Sigma}_k^t = \left(\left(\boldsymbol{\Sigma}_{k-1}^t\right)^{-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}\boldsymbol{W}_1\right)^{-1}. \quad (24)$$

The above update equations are identical to Kalman filtering update Kalman (1960). Hence, the state is updated once for every pair of measurements received, until a convergence condition is reached, or maximum number of updates is reached.

4.1.3. Simultaneous Multi-measurement Update. So far we have considered only the case where the state is updated once per pair of measurements. However, such an approach can be inefficient when applied to pose estimation from stereo cameras or Kinect™. In such applications, one typically obtains several position measurements at each time instant and processing the measurements in a pairwise manner can be time consuming. In order to address this situation, we can rewrite Eq. 13 as:

$$\boldsymbol{H}_j\widetilde{\boldsymbol{q}} = 0, \quad j = 1, \ldots, m.$$

$\boldsymbol{H}_j$ has the form as shown in Eq. 14, where $\boldsymbol{a}_v$, $\boldsymbol{b}_v$ are obtained from point-pairs constructed by subtracting random pairs of points or subtracting each point from the centroid (similar to Horn (1987)). Since the measurements are

assumed to be independent, we have

$$p(z_k|\widetilde{q}_k) = \prod_{j=1}^{m} \frac{1}{N_2^j} \exp\left(\frac{-1}{2}(H_j\widetilde{q}_k)^T \widetilde{q}_k^{-1}(H_j\widetilde{q}_k)\right),$$

$$= \frac{1}{N_3} \exp(\widetilde{q}_k^T D_3 \widetilde{q}_k), \qquad (25)$$

where $D_3 = \frac{1}{2}\sum_j \left(-H_j^T \widetilde{q}_k^{-1} H_j\right)$ and $N_3 = \prod_{j=1}^{m} N_2^j$. Eq. 21 can be rewritten as

$$p(\widetilde{q}_k|z_k) \propto \frac{1}{N_1} \exp\left(\widetilde{q}_k^T D_1 \widetilde{q}_k\right) \frac{1}{N_3} \exp\left(\widetilde{q}_k^T D_3 \widetilde{q}_k\right)$$

$$\propto \exp\left(\widetilde{q}_k^T M_k Z_k M_k^T \widetilde{q}_k\right), \qquad (26)$$

where $M_k Z_k M_k^T$ is obtained from Bingham multiplication. $\widetilde{q}_k$ and t_k are obtained as shown in Sec. 4.1.2.

4.2. Surface-normal Measurements

In some applications, in addition to position measurements, surface-normal measurements may also be available (Billings et al. (2015); Srivatsan et al., (2016a)). The following equation relates the surface-normals in the two frames,

$$\widetilde{n}_i^a = \widetilde{q} \odot \widetilde{n}_i^b \odot \widetilde{q}^* \quad i = 1,\dots,l$$

$$\Rightarrow \widetilde{n}_i^a \odot \widetilde{q} = \widetilde{q} \odot \widetilde{n}_i^b$$

$$\Rightarrow J_i \widetilde{q} = 0, \quad \text{where}$$

$$J_i = \begin{bmatrix} 0 & -(n_i^a - n_i^b)^T \\ (n_i^a - n_i^b) & (n_i^a + n_i^b)^\times \end{bmatrix},$$

where n_i^a are surface-normals in CAD-model frame and n_i^b are surface-normals in the sensor frame. Similar to the derivation in the case of position measurements (see Eq. 25), we obtain,

$$p(z_k|\widetilde{q}_k) = \frac{1}{N_4} \exp(\widetilde{q}_k^T D_4 \widetilde{q}_k), \qquad (27)$$

where $D_4 = \frac{1}{2}\sum_i \left(-J_i^T S_k^{-1} J_i\right) + \frac{1}{2}\sum_j \left(-H_j^T \widetilde{q}^{-1} H_j\right)$, S_k is the pseudo-measurement uncertainty. Thus, we have

$$p(\widetilde{q}_k|z_k) \propto \frac{1}{N_1} \exp(\widetilde{q}_k^T D_1 \widetilde{q}_k) \frac{1}{N_4} \exp(\widetilde{q}_k^T D_4 \widetilde{q}_k),$$

$$\propto \exp(\widetilde{q}_k^T M_k Z_k M_k^T \widetilde{q}_k).$$

Instead of using a Gaussian distribution to model the uncertainty in the surface-normal measurements, a von Mises Fisher distribution may be chosen, as shown by Billings et al. (2015). For calculating the uncertainties using Proposition 1, the covariance of the von Mises distribution can be calculated as shown by Hillen et al., (2017).

4.3. Pose Measurements

Systems that use pose measurements for model update typically have the following general form (Park and Martin (1994); Ackerman et al. (2013))

$$AX - XB = 0, \qquad (28)$$

where $A, X, B \in SE(3)$. These problems are generally referred to as 'hand-eye calibration'. A and B are pose-measurements and X is the desired transformation to be estimated. While variants of this problem exist in the form of $AX = BY$ (also known as hand-eye robot-world calibration) (Dornaika and Horaud (1998); Zhuang et al., (1994)), these problems can also be reduced to the form of $AX = XB$ by using relative measurements (Tabb and Yousef (2015)). Let $\widetilde{a}^r, \widetilde{b}^r \in \mathbb{R}^4$ be the unit quaternion and $a^t, b^t \in \mathbb{R}^3$ be the translation, parameterizing A and B respectively. For Eq. 28, we have

$$\widetilde{a}^r \odot \widetilde{q} - \widetilde{q} \odot \widetilde{b}^r = 0, \quad \text{and,} \qquad (29)$$

$$\widetilde{a}^t \odot \widetilde{a}^r \odot \widetilde{q} + \widetilde{a}^r \odot \widetilde{t} \odot \widetilde{q}$$

$$-\widetilde{t} \odot \widetilde{q} \odot \widetilde{b}^r - \widetilde{q} \odot \widetilde{b}^t \odot \widetilde{b}^r = 0, \qquad (30)$$

where $\widetilde{a}^t \triangleq (0,(a^t)^T)^T$, $\widetilde{b}^t \triangleq (0,(b^t)^T)^T$. From Eq. 29 and Eq. 1, we obtain

$$L^r(\widetilde{a}^r, \widetilde{b}^r)\widetilde{q} = 0, \quad \text{where,} \qquad (31)$$

$$L^r(\widetilde{a}^r, \widetilde{b}^r) = F_1(\widetilde{a}^r) - F_2(\widetilde{b}^r) \in \mathbb{R}^{4\times4}.$$

Eq. 31 is linear in $\widetilde{q}$, as was the case in the previous sections. Thus we follow a similar analyses to estimate $\widetilde{q}_k$ and the associated uncertainty.

Once we obtain $\widetilde{q}_k$, its value is substituted in Eq. 30 and the terms are rearranged to obtain

$$L^t(\widetilde{a}^r, a^t, \widetilde{b}^r, b^t)\widetilde{t} + \widetilde{\tau}_3 = 0, \quad \text{where,} \qquad (32)$$

$$L^t = F_1(\widetilde{a}^r) - F_2(\widetilde{q}_k \odot \widetilde{b}^r \odot \widetilde{q}_k^*),$$

$$\widetilde{\tau}_3 = \widetilde{a}^t \odot \widetilde{a}^r - \widetilde{q}_k \odot \widetilde{b}^t \odot \widetilde{b}^r \odot \widetilde{q}_k^*.$$

Eq. 32 can be simplified as shown

$$\overline{L^t} t + \tau_3 = 0,$$

where $\overline{L^t}$ is a matrix composed of the last three columns of L^t.

We use a maximum a priori estimate to obtain t_k and Σ_k^t similar to Eq. 23 and Eq. 24,

$$t_k = \left(\left(\Sigma_{k-1}^t\right)^{-1} + \overline{L^t}^T R_k^{-1} \overline{L^t}\right)^{-1}$$

$$\left(\left(\Sigma_{k-1}^t\right)^{-1} t_{k-1} - (L^t)^T R_k^{-1} \tau_3\right), \qquad (33)$$

$$\Sigma_k^t = \left(\left(\Sigma_{k-1}^t\right)^{-1} + \overline{L^t}^T R_k^{-1} \overline{L^t}\right)^{-1}, \qquad (34)$$

where the uncertainty R_k can be obtained following steps similar to those shown in Sec. 4.1.2.

4.4. Dealing with Unknown Data-association

In this section, we discuss the general approach we follow for pose estimation when the data association between sensor point and the model is unknown. We explain the approach for the example of position measurements, but

the general idea can be easily extended for other forms of measurements.

Let ψ be the model shape that is often available in the form of a triangulated mesh. Let $\boldsymbol{b}_i^s \in \mathbb{R}^3$ be the sensor measurements in the sensor's reference frame. If we knew the point on ψ that is associated with $\boldsymbol{b}_i^s$, then we could follow the approach as shown in Sec. 4.1. However, we do not know this data association, and so we find the point $\boldsymbol{a}_i \in \psi$, such that

$$\boldsymbol{a}_i = \underset{\boldsymbol{a} \in \psi}{\operatorname{argmin}} |\boldsymbol{a} - Rot(\widetilde{\boldsymbol{q}}_k)\,\boldsymbol{b}_i^s - \boldsymbol{t}_k|. \qquad (35)$$

This approach is referred to as 'closest point' correspondence and is often used in methods such as ICP (Besl and McKay (1992)). An alternate approach involves using a probabilistic criteria ('most likely' correspondence) instead of Eq. 35,

$$\boldsymbol{a}_i = \underset{\boldsymbol{a} \in \psi}{\operatorname{argmax}} \frac{1}{N} \exp\left(-\frac{1}{2}\boldsymbol{v}_k^T \boldsymbol{S}_k \boldsymbol{v}_k\right), \qquad (36)$$

$$= \underset{\boldsymbol{a} \in \psi}{\operatorname{argmin}} \boldsymbol{v}_k^T \boldsymbol{S}_k \boldsymbol{v}_k, \qquad (37)$$

where $\boldsymbol{v}_k = \boldsymbol{a} - Rot(\widetilde{\boldsymbol{q}}_k)\,\boldsymbol{b}_i^s - \boldsymbol{t}_k$, and $\boldsymbol{S}_k = Rot(\widetilde{\boldsymbol{q}}_k)\,\boldsymbol{\Sigma}^{\boldsymbol{b}_i^s}Rot(\widetilde{\boldsymbol{q}}_k)^T + \boldsymbol{\Sigma}^{\boldsymbol{a}}$ (Billings et al. (2015)). Eq. 36 does not take into account the uncertainty associated with $\widetilde{\boldsymbol{q}}_k$ and $\boldsymbol{t}_k$. While it is straightforward to estimate the uncertainty in the pose and incorporate in Eq. 36, we suspect that doing so can result in worse data associations, especially when the initial pose-uncertainty is high. Additionally, both the 'closest point' as well as 'most likely' paradigms are equivalent when the uncertainties $\boldsymbol{\Sigma}^{\boldsymbol{b}_i}, \boldsymbol{\Sigma}^{\boldsymbol{a}}$ are isotropic (Billings et al. (2015)).

4.4.1. k-d Tree Search: A naive implementation for optimizing Eq. 35 or Eq. 36 would require checking every point in ψ and find the one that minimizes the objective. However, such an approach takes $O(n)$ time on an average, which can be practically infeasible in many situations. As a result a popular approach to search for the optimal point in the model involves using a k-d tree (Friedman et al., (1977)) (see Fig. 5(b)). A k-d tree with n points takes $O(\log n)$ time on an average per search. There are different ways to build k-d trees depending on which coordinate is chosen for splitting the data. One often chooses the coordinate with the largest spread. One variant is the dyadic tree which cycles through the coordinates and splits the data at the midpoint. There are several other variants of the k-d tree such as ball tree (Omohundro (1989)), Vantage-point tree (Yianilos (1993)), etc.

4.4.2. Principal Direction Tree Search: One may also use a principal direction (PD) tree (Verma et al. (2009)). The primary difference between the k-d and PD trees is that each node of the PD tree has a local coordinate system that is oriented based on the spread of the points in ψ instead of
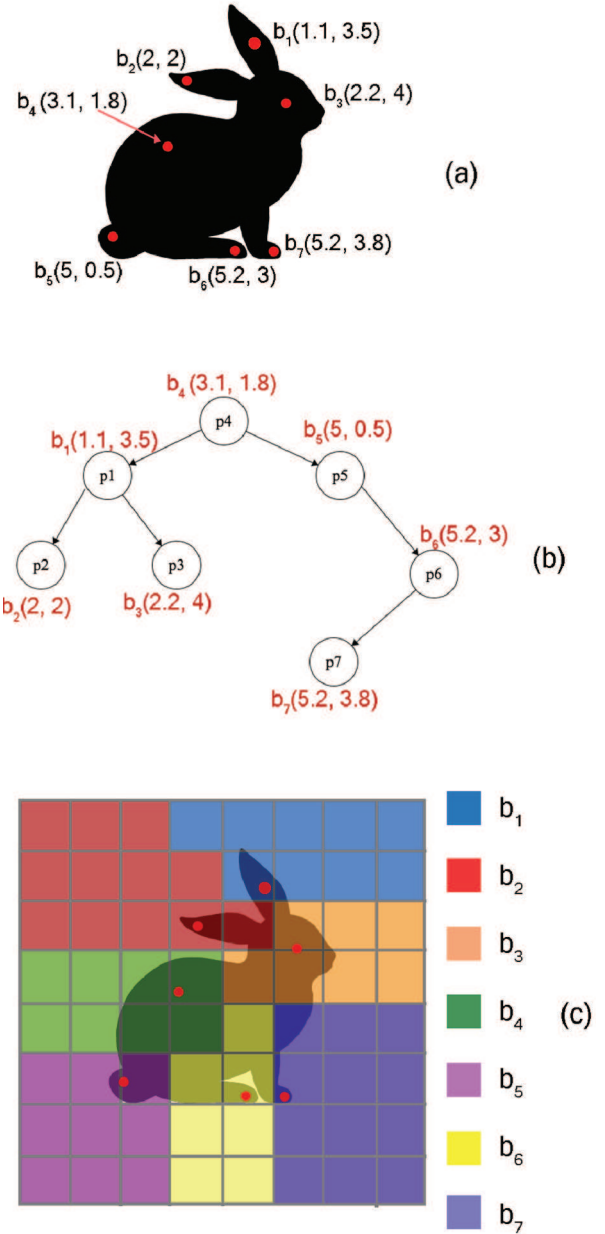

Fig. 5. (a) Shows a representative point cloud of a model. For ease of demonstration, we show a 2D case. Let the model have 7 points as shown by red dots. (b) k-d tree constructed for the model points. (c) Look-up table approach is illustrated. A grid is created around the model. The closest point from each grid center to the model is computed using k-d tree. Each grid-color shows an index corresponding to the closest point on the model. Given a new sensor point, we first transform it to the model frame (using the best estimate of transformation). We then find which grid it belongs to and return the precomputed closest point.

being axis-aligned with the model's reference frame. Such a tree can result in more compact geometric bounds of nodes within the PD tree and hence provide a boost to the search efficiency. Billings et al. (2015) use a PD tree search in their

algorithm and demonstrate improved performance over k-d tree search. A variant of this method is the random projection tree (RP tree) (Dasgupta and Freund (2008)). In this tree a random vector is chosen and the tree is split along the median of the data projected onto this vector. The RP tree is computationally faster than PD tree but produces lower accuracy results. McCartin-Lim et al., (2012) have developed an approximate PD tree (APD tree) that has the computational complexity of an RP tree with the accuracy comparable to a PD tree.

4.4.3. Fast Look-up Table Search: In this work, we have developed a faster (albeit more approximate) approach to search for the closest point to the model. We refer to this approach as 'look-up table' approach. As shown in Fig. 5(c), we use a uniform grid to discretize the space around the model. We then compute the closest point from the center of each grid to the model using a k-d tree. The index of the closest point on the model to each grid point g_i is stored in a look-up table. Such a look-up table needs to be constructed once before the start of the experiments. When we obtain a sensor measurement b_i, we find the closest grid-center to $\left(Rot(\widetilde{q}_k)\, b_i + t_k\right)$, and return the corresponding pre-computed closest point. Since the grid is uniform, the computation of the closest grid center can be done in $O(1)$.
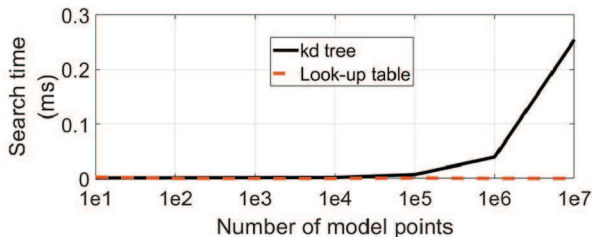

Fig. 6. Plot shows the time taken per search vs number of model points for k-d tree and look up table approach. Irrespective of the number of model points, the search time with look-up table approach remains of the same order ($\approx 2 \times 10^{-3}$ ms). k-d tree, on the other hand, takes 10^{-1} ms for 10 model points (which is 2 times faster than look-up table) and 0.25 ms for 1 million model points (which is 100 times slower than look-up table).

We choose a uniform grid in a box, which is three times the largest dimension of the model, with 100 points in each dimension. The center of the grid space is chosen to coincide with the center of the model. An advantage of this search method is that the closest point computation is carried offline, and finding the closest grid-center is an inexpensive operation. Fig. 6 shows that when the model has $> 10,000$ points, the look-up table approach outperforms k-d tree by several orders of magnitude[1]. However, a shortcoming of this approach is that the density of the grid dictates the accuracy of the result, because every point inside each grid is assigned the same closest point. Increasing the density of the grid would result in more accurate

results at the cost of increase memory storage to save the lookup table.

It is interesting to note that Fig. 5 is actually a grid approximation of the Voronoi diagram of the model points. The grid approximation allows for quick retrieval of closest point, as opposed to saving the Voronoi cells and checking for which Voronoi cell a given point belongs to.

5. Results

In this section, we consider two scenarios for using the Bingham distribution-based linear filter: (1) known data association and (2) unknown data association. Without loss of generality, we choose the following values for all experiments, $M_0 = I_{4\times4}$, $Z_0 = diag(0, -1, -1, -1) \times 10^{-300}$ which represents an uninformative prior with high initial uncertainty.

5.1. Known data association

5.1.1. Simulation example: point cloud registration. In this section, we assume that the correspondence between points $a_i \in \mathbb{R}^3$ and $b_i \in \mathbb{R}^3$ are known, and estimate the pose between the frames that these two point sets lie in.

The coordinates of the dataset a_i^s are produced by drawing 100 points uniformly in the interval $[-250 \text{ mm}, 250 \text{ mm}]$. To create the noiseless dataset b_i, a random transformation is applied to a_i. This transformation is generated by uniformly drawing the rotational and translational parameters in the intervals $[-180°, 180°]$ and $[-100 \text{ mm}, 100 \text{ mm}]$, respectively. In Experiment 1, no noise is added to b_i. In Experiment 2 and Experiment 3, a noise uniformly drawn from $[-2 \text{ mm}, 2 \text{ mm}]$ and $[-10 \text{ mm}, 10 \text{ mm}]$, respectively, is added to each coordinate of b_i. In Experiment 4 and Experiment 5, a Gaussian noise drawn from $\mathcal{N}(0, diag(2, 2, 2))$ and $\mathcal{N}(0, diag(10, 10, 10))$, respectively, is added to each b_i.

The linear Bingham filter (BF) is used to estimate the pose in each of the experiments in a sequential manner. This procedure is repeated 1000 times with different datasets and different transformations that are randomly generated. The results are compared with dual quaternion filter (DQF) (Srivatsan et al., 2016), Moghari-UKF (Moghari and Abolmaesumi, 2007), and Pennec-EKF (Pennec and Thirion, 1997). We also compare the results with an incremental variant of Horn's method (Horn-Inc) (Horn, 1987). In this method we find estimate the pose once for every four measurements obtained using the estimate from the previous update as an initial condition.

For each of the methods, if the root-mean-square (RMS) error at the end of any experiment is greater than 250 mm (the size of the workspace considered), then we consider the experiment to have failed. The average RMS errors

---

1. The time reported is for a script written in MATLAB R2017b software from MathWorks, running on a ThinkPad T450s computer with 8 GB RAM and intel i7 processor.
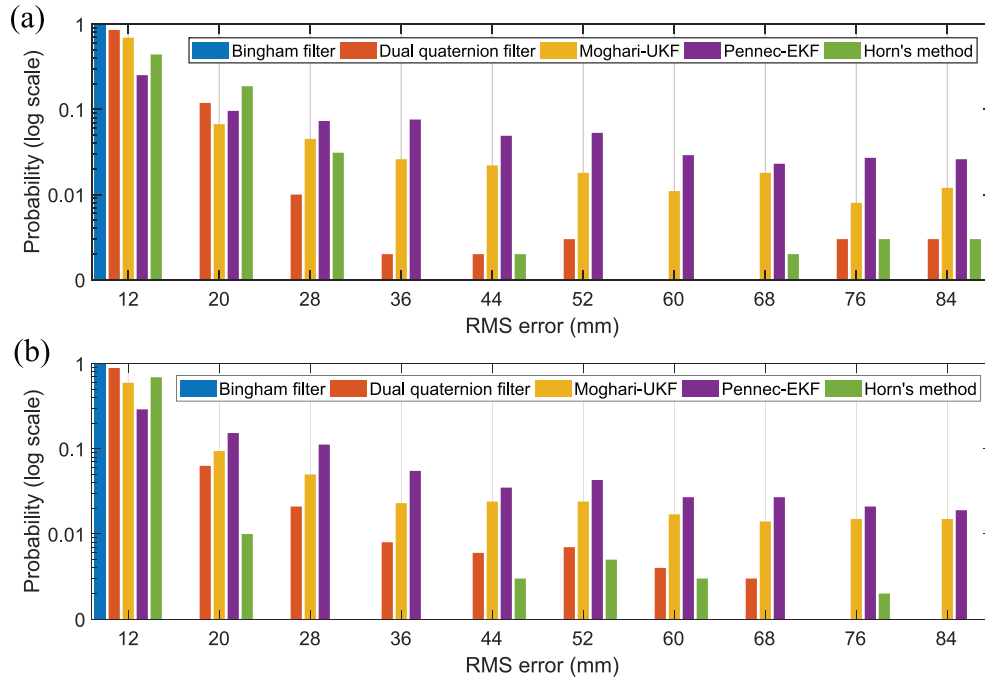
Fig. 7. Histogram shows the RMS errors for the Bingham filter (BF), dual quaternion filter of Srivatsan et al. (2016) (DQF), UKF of Moghari and Abolmaesumi (2007) (Moghari-UKF), EKF of Pennec and Thirion (1997) (EKF) and an incremental variant of Horn's method (Horn, 1987). (a) The results shown are for Experiment 3, where the sensed points have a noise uniformly drawn from [−10 mm, 10 mm]. The BF is most accurate with an average RMS error of 10.30 mm and a 100% success rate. (b) The results shown are for Experiment 5, where the sensed points have a noise obtained from a Gaussian distribution $\mathcal{N}(\mathbf{0}, diag(10, 10, 10))$. The BF is the most accurate with an average RMS error of 4.95 mm and a 100% success rate.

Table 1. Mean RMS errors for experiments involving three different levels of uniform measurement noise.

	Experiment 1		Experiment 2		Experiment 3	
	RMS (mm)	Success (%)	RMS (mm)	Success (%)	RMS (mm)	Success (%)
Our approach	0.00	100	2.06	100	10.30	100
DQF	0.00	100	2.72	99.70	12.17	99.90
Horn-Inc	0.00	100	4.96	97.30	34.80	78.30
Moghari-UKF	15.97	99.50	18.84	99.30	21.00	99.50
Pennec-EKF	74.20	88.20	83.41	83.90	47.312	94.80

Table 2. Mean RMS errors for experiments involving two different levels of Gaussian noise in the measurements.

	Experiment 4		Experiment 5	
	RMS (mm)	Success (%)	RMS (mm)	Success (%)
Our approach	0.210	100.00	4.947	100.00
DQF	0.214	100.00	6.980	99.80
Horn-Inc	0.310	100.00	6.651	99.20
Moghari-UKF	15.960	99.40	20.770	99.70
Pennec-EKF	40.480	95.50	44.188	93.90

over the first three experiments along with the percentage of successful runs are listed in Table 1. Note that we do not include failed experiments in our computation of RMS errors.

Figure 7(a) shows the histogram of errors for Experiment 3. The BF always estimates the pose with the lowest RMS error. The RMS error of DQF and Moghari-UKF are both small, but larger than the BF. Owing to the large initial orientation chosen, they get trapped in local minima sometimes, which is captured by the bars at higher RMS errors in Figure 7(a). Whereas Horn-Inc performs better than Moghari-UKF when the noise is small, higher noise

results in the Horn-Inc performing worse. Pennec-EKF performed the worst in all three experiments. For Experiment 3, the average run time for the BF is 26 ms, compared with 9 ms of DQF, 3.8 ms of Horn-Inc, 130 ms of Moghari-UKF, and 17 ms of Pennec-EKF.

The average RMS errors for Experiment 4 and Experiment 5 along with the percentage of successful runs are tabulated in Table 2.

Figure 7(b) shows the histogram of errors for Experiment 3. Although the trend is similar to the case of the first three experiments, all the approaches perform better than in the case of uniform noise. This is an expected behavior, because the filters are developed to handle Gaussian noise in the
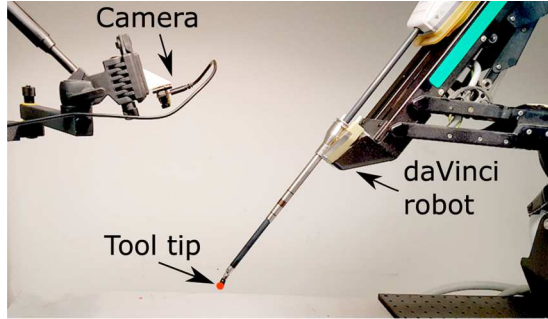
Fig. 8. A spherical tool tip is attached to the da Vinci robot. The tip is tracked using a stereo camera, which is held in a fixed position. As the robot is telemanipulated, the spherical tool-tip is tracked using the stereo camera, and the relative pose between the camera frame and the robot frame is estimated.

measurements. BF once again is most accurate among all the methods.

5.1.2. Real-world example: registering camera and robot frame. Figure 8 shows an arm of a da Vinci® surgical robot (Intuitive Surgical Inc., Mountain View, CA) mounted on a table, and a stereo camera (ELP-1MP2CAM001 Dual Lens) mounted on a rigid stand. The relative pose between the robot's frame and the camera's frame is fixed, and needs to be estimated. To estimate this pose, the robot is telemanipulated in arbitrary paths and the location of tip of the robot a_i is computed in the camera frame by segmenting the tip from the stereo image and estimating its center. The position of the tip in the robot frame, b_i is obtained from the kinematics of the robot. The pose between the points a_i and b_i can be obtained as shown in Section 5.1.

Table 3 shows the RMS error and the time taken for estimation by the Bingham filtering (BF) approach using pairwise updates, using 20 simultaneous measurement-pairs per update (abbreviated as BFM-20 in the table), and an incremental Horn's method (Horn-Inc) where the pose is estimated once per 20 measurements. We evaluate the results for three termination criteria.

1. Criterion 1: If the change in the translation and rotation fall below a set threshold, the algorithm is terminated. In this example we use 1 mm and $0.1°$ as threshold for translation and rotation, respectively.
2. Criterion 2: If the conditions of Criteria 1 are satisfied and the uncertainty in the rotation and translation fall below a set threshold. In this example, we choose the threshold for uncertainty in translation as 10^{-4} for the largest eigenvalue of Σ_k^t and -10^3 for the largest non-zero diagonal element of Z_k.
3. Criterion 3: The algorithms are terminated after processing a set number of measurements. In this example we choose this number $n = 200$.

We observe that BFM-20 produces more accurate and faster results than the BF, because multiple simultaneous

measurements help smooth out the effect of the noise in the measurements. As Horn-Inc does not contain any uncertainty information, results of Criteria 1 and 2 are identical. Uncertainty update in BFM-20, however, prevents premature convergence due to Criterion 1 and results in more accurate estimate with Criterion 2. Instead of converging after using 28 data points, BFM-20 takes 31 points to converge according to Criterion 2.

When all three algorithms are run for 200 measurements, the accuracy of BFM-20 is the highest. Unlike BF and BFM-20, after processing 200 points, Horn-Inc has a higher RMS error. This is because in the filtering approaches, information from previous measurements are "baked" into the current estimate by the uncertainty update, which is absent in Horn-Inc. Overall, Horn-Inc takes lowest computation time and performs better than BF, but produces higher errors than BFM-20. We also perform a batch optimization using all 200 points and Horn's method (abbreviated as Horn-Batch in the table), as well as Bingham filter with a single update using 200 point measurements (abbreviated as BF-200). Both BFM-200 and Horn-Batch converge to the same RMS error and take comparable computation time, with the BFM-200 taking 0.14 ms longer due to additional computations involving uncertainty update.

We also implement a naive outlier detector for all the algorithms, except BF. For every mini-batch of measurements used per update, we first apply the current estimate of the pose and sort the measurements b_i based on their proximity to a_i. We then pick a fraction of the measurement pairs that are closest and discard the rest (we choose a fraction of 0.6 in this work). We update using only this fraction of measurements. Any outlier present in the current batch of measurements would be discarded in this process.

As we did not observe many outliers in the experiment, we created a simulated example to test our approach in a more challenging case. In this simulation experiment, we take the measurements from the robot experiment described above and randomly chose 16% of the points to be outliers. We then estimate the pose using BFM-20 and Horn-Inc as shown in Figure 9(b). In the absence of outlier detection both BFM-20 and Horn-Inc are highly inaccurate due to the presence of outliers. Upon using outlier detection, both methods perform well, with the BFM-20 producing smoother estimates with lower RMS error than Horn-Inc as shown in Table 4.

5.2. Unknown data association

5.2.1. Simulation example: point-cloud registration. In this section we assume that the points a_i and surface normals n_i^a are the vertices and normals, respectively, of a triangulated mesh. Figure 10 shows the triangulated mesh in the shape of a bunny (Turk and Levoy, 2005), which has 86,632 triangles.

Table 3. Experimental results for robot–camera registration.

	Criteria 1			Criteria 2			Criteria 3		
	Time (ms)	RMS (mm)	n	Time (ms)	RMS (mm)	n	Time (ms)	RMS (mm)	n
BF	5.81	6.11	53	5.89	6.11	53	25.29	3.34	200
BFM-20	1.53	2.81	28	1.66	2.77	31	29.81	2.74	200
Horn-Inc	0.43	2.84	25	0.43	2.84	25	14.16	2.86	200
BFM-200	—	—	—	—	—	—	0.67	2.69	200
Horn-Batch	—	—	—	-	—	—	0.53	2.69	200

Table 4. Simulation results for robot–camera registration.

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	Time (ms)	RMS (mm)
Actual	11.79	261.49	27.29	178.25	7.29	−130.46	—	—
BFM-20	11.78	261.45	27.52	178.38	7.33	−130.35	42	2.79
Horn-Inc	20.67	259.21	27.52	177.08	5.73	−131.42	34	3.66
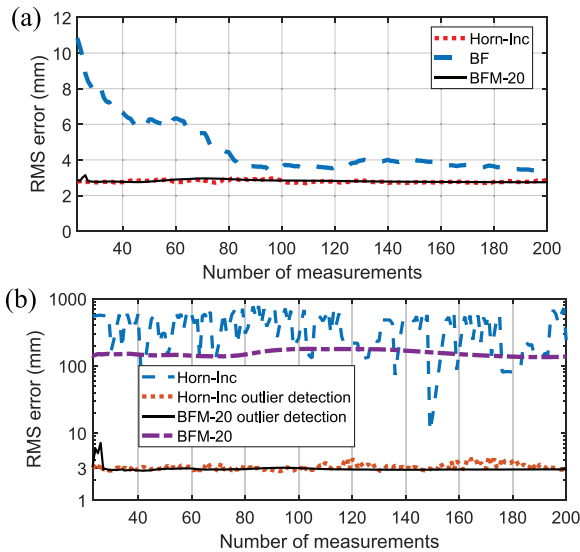
Fig. 9. (a) Experimental results for robot–camera registration. BFM-20 produces the most accurate pose estimation, whereas Horn-Inc is the fastest, and BF is the most erroneous. (b) Robot–camera registration results in the presence of 16% measurement outliers. In the absence of outlier detection both BFM-20 and Horn-Inc perform poorly. Outlier detection improves both BFM-20 and Horn-Inc, with the BFM-20 being smoother and more accurate.

We randomly pick 5,000 points from the triangulated mesh and to each coordinate of the points, add a noise uniformly drawn from $[-2\,\text{mm}, 2\,\text{mm}]$. For each $(\boldsymbol{b}_i, \boldsymbol{n}_i^b)$, the correspondence is obtained by finding the closest point-normal pair $(\boldsymbol{a}_i, \boldsymbol{n}_i^a)$ on the triangulated mesh.

We estimate the pose using the BF with 20 simultaneous multi-measurements as described in Section 4.1.3. Figure 11 shows the RMS error vs number of simultaneous measurements used. Update based on one pair of measurements
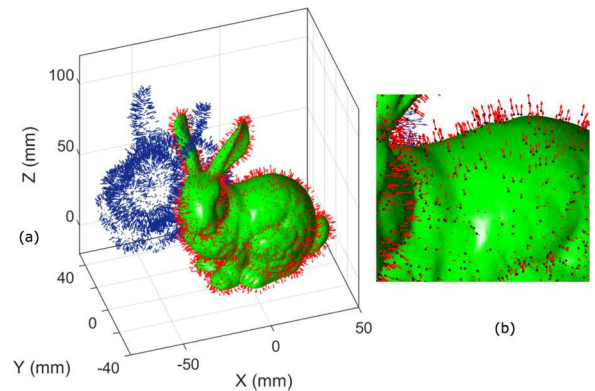

Fig. 10. (a) Triangulated mesh of Stanford bunny (Turk and Levoy, 2005) is shown in green. Blue arrows represent initial location and red arrows represent estimated location of points and surface normals. (b) Enlarged view shows that the estimated location of points accurately rests on the triangulated mesh and the estimated direction of the surface normals aligns well with the local surface normal. The Bingham filter takes 2.4 s in Matlab and 0.08 s in C++ to estimate the pose.
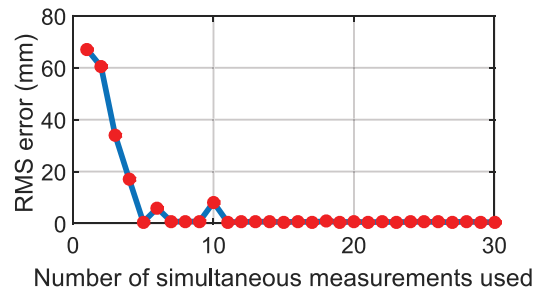

Fig. 11. RMS error upon convergence versus number of simultaneous measurements used. The greater the number of simultaneous measurements used, the lower the RMS error.
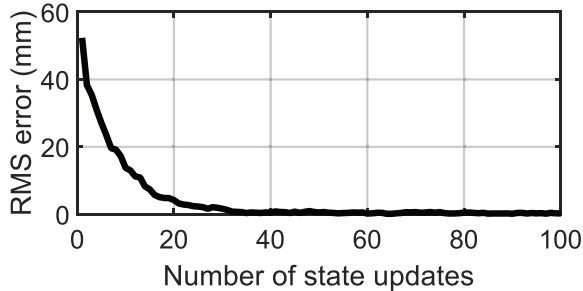
Fig. 12. RMS error in the pose versus number of state updates as estimated by the Bingham filter using 20 simultaneous position and normal measurements in each update. The estimate converges in around 40 iterations.

results in a local optimum (RMS error is $\approx$ 70 mm as shown in Figure 11). However, the performance drastically improves when > 10 simultaneous measurements are used.

The penultimate row of Table 5 shows the pose parameters as estimated by the BFM-20. We also estimate the pose using 20 simultaneous surface-normal and position measurements (abbreviated as BFN-20 in Table 5). The RMS error for the BFN-20 is lower than the BFM-20, but the time taken is higher because of additional computations in the correspondence step involving surface normals. Figure 10(a) shows the initial position of the surface normals and point locations with blue arrows and the BFN-20 estimated surface normals and point locations with a red arrow. The enlarged view in Figure 10(b) shows that our approach accurately registers the points as well as aligns the surface normals to the triangulated mesh.

Table 5 also shows the pose parameters as estimated by ICP (Besl and McKay, 1992), DQF (Srivatsan and Choset, 2016), Pulli's method (Pulli, 1999), Go-ICP (Yang et al., 2013), and IMLOP (Billings and Taylor, 2014). For the sake of a fair comparison, we use k-d tree search for the correspondence in all the methods except IMLOP. The correspondence criteria of IMLOP does not allow the use of a k-d tree and the authors use a specialized PD tree search. Go-ICP takes 0.8 s for estimating the pose parameters. However, we do not report the time taken for Go-ICP in Table 5 as the code originally supplied by the authors runs in C++ whereas all other algorithms run on Matlab. BFM-20 implemented in Matlab takes less time than Go-ICP and is more accurate as well. This improvement in accuracy is attributed to the fact that Go-ICP does not consider uncertainties in measurements, whereas our approach does.

BFM-20 and BFN-20 are accurate and orders of magnitude faster than all other methods. Figure 12 shows the RMS error at the end of each update step for BFN-20. The RMS error reduces to < 0.6 mm at around 40 state updates. To obtain the same accuracy as DQF and ICP ($\approx$ 2 mm), both BFM-20 and BFN-20 take $\approx$ 30 state updates, which takes 0.28 s. The accuracy of Pulli is greater than ICP because it uses surface-normal information to prune the

correspondence choices, which greatly helps with the registration. Whereas Go-ICP does not use any surface-normal information, it performs a global search and, hence, produces results that are more accurate than ICP and Pulli. IMLOP uses the point and surface-normal information and is as accurate as BFN-20 but takes several orders of magnitude more computation time. Furthermore, BFN-20 and BFM-20 produce serial updates and can be terminated using a criteria as described in Section 5.1.2. This is not possible in the case of IMLOP, which is a batch processing method and uses all the measurements to produce the pose estimate.

5.2.2. Simulation example: comparing k-d tree versus look-up table. In this section, we evaluate the accuracy and the time taken for registering points to a model using k-d tree and look-up table-based approach. For this we consider the Lucy model from Stanford point cloud library dataset (Turk and Levoy, 2005). The model has 1.2 million points. We first scale the model so that it fits within a cube of size 1 unit. We randomly sample 2,500 points from this model and add a noise to it that is sampled from $\mathcal{N}(\mathbf{0}, diag(4, 4, 4) \times 10^{-4})$. We then apply a known transformation to these points and then try to estimate that transformation using the BFM with 20 simultaneous measurements. The transformation applied is $(0.01, -0.02, 0.05)$ units in translation and $(10, -10, 25)°$ in rotation about each axis. We evaluate the performance of our approach using k-d tree as well as look-up table with varying grid sizes. The results are shown in Table 6. Note that the time reported does not include the time taken to construct the k-d tree or the look-up table.[2] As expected the k-d tree performs the best in terms of accuracy. The look-up table is several magnitudes of order faster. As we increase the grid density the accuracy of the look-up table increases as well, whereas search time remains constant.

5.2.3. Real-world example: point-cloud stitching. Stereo imaging devices such as the Microsoft Kinect™ offer colored point cloud data (RGB-D: color and depth data), which is generated using a structured light-based depth sensor. The Kinect™ is widely used in robot navigation (Khoshelham and Elberink, 2012) and object manipulation (Engelhard et al., 2011). In this work, we align a pair of point cloud data obtained from the Kinect™, using the Bingham filter, to develop a point-cloud model of the environment. It is assumed that there is some overlap between the two point clouds. We demonstrate our approach on RGB-D images taken from the "Freiburg1-Teddy" dataset of Sturm et al. (2012). Figure 13(a) and (b) show the snapshots of the images. Figure 13(d) shows the final model of the room as generated by our approach. We use 20 simultaneous measurements and the same initial conditions as in the previous cases. Our approach takes $\approx$ 0.21 s for estimating pose, which is twice as fast as ICP which takes $\approx$ 0.46 s. To improve the speed we have implemented a C++ version of the Bingham filter, which takes only $\approx$ 2 ms.[3] The RMS

Table 5. Results for registration of points and surface normals to the geometric model.

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	Time (s)	RMS (mm)
Actual	44.83	−50.45	7.15	−12.01	−21.49	−28.14	—	—
DQF	44.83	−50.44	7.14	−12.02	−21.48	−28.14	13.21	1.89
ICP	44.52	−49.16	6.32	−9.05	−19.11	−30.40	77.83	2.04
Pulli	44.79	−49.83	6.83	−10.56	−20.50	−29.60	10.23	0.90
Go-ICP	44.69	−50.49	7.86	−12.04	−21.58	−27.83	—	0.72
IMLOP	44.91	−50.39	7.58	−12.01	−21.62	−28.11	472.8	0.32
BFM-20	44.44	−50.60	7.18	−12.66	−21.80	−27.58	0.38	0.41
BFN-20	44.60	−50.59	7.58	−12.01	−21.69	−28.29	2.46	0.32

Table 6. RMS error and time taken to register points to a geometric model with 1.2 million points.

Approach	Grid points	Time (s)	RMS (mm)
*k*d tree	—	2.62	0.013
Look-up table	$20 \times 20 \times 20$	0.004	0.046
Look-up table	$40 \times 40 \times 40$	0.004	0.029
Look-up table	$100 \times 100 \times 100$	0.004	0.021
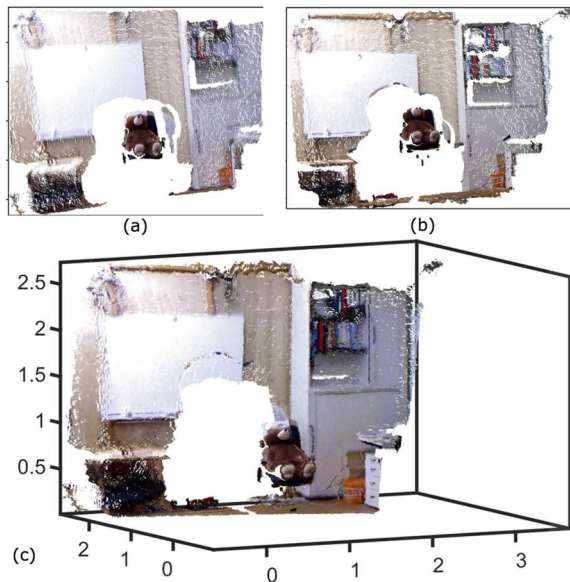
Fig. 13. (a), (b) RGB-D images obtained from Kinect™, with some overlapping region. (c) The point cloud model estimated by aligning the point clouds in (a) and (b) using the Bingham filter. The Bingham filter takes 0.21 s to estimate the pose with a RMS error of 4.4 cm, as opposed to ICP, which takes 0.46 s with a RMS error of 6 cm.

error of our approach is 4.4 cm, which is of the order of the accuracy of the sensor itself (Khoshelham and Elberink, 2012) and is better than the RMS error of ICP of 6 cm.

5.2.4. Real-world example: stereo point cloud registration and tracking. In this section we look at an example of stereo image registration. The transformation between

camera-frame and model-frame is estimated by registering the reconstructed point cloud from stereo images with the geometric model of the object.

As the stereo camera images can consist of a number of objects, we first manually select the region containing the object of interest. We then refine the selection using a graph cut-based image segmentation. We use the variant of the BF that uses multiple simultaneous point measurements to estimate the pose. As the pose estimation is fast, we repeatedly estimate the pose even after convergence. This allows us to track the object in the frame of the camera subject to slow motions. We use this approach to track three objects with varying level of geometric complexity: a Stanford bunny, a pelvis bone, and a prostate. The bunny is geometrically most expressive and easy to track. The prostate is relatively symmetric and lacks interesting geometric features and is the most difficult of the three objects to track.

Figure 14 shows the RMS error over a 60 s time period of tracking, in which the bunny was moved by a user in arbitrary fashion. In the first 15 s, the user provides a region of interest in the stereo images and segments the object based on its hue and saturation. The BF then estimates the pose. Note that between 15 and 50 s, the bunny is being manually moved and our approach is able to robustly and accurately track the pose (for example, instances A and B in Figure 14). The RMS error is < 2 mm, most of which can be attributed to the noise in the stereo point cloud. After about 55 s, the filter produces erroneous pose estimates (for example, instance C in Figure 14). This is because the bunny was moved very quickly and the incremental errors in the pose estimate resulted in a large difference from one frame to another and caused the point cloud segmentation to lose track of which points it was meant to be registering to.
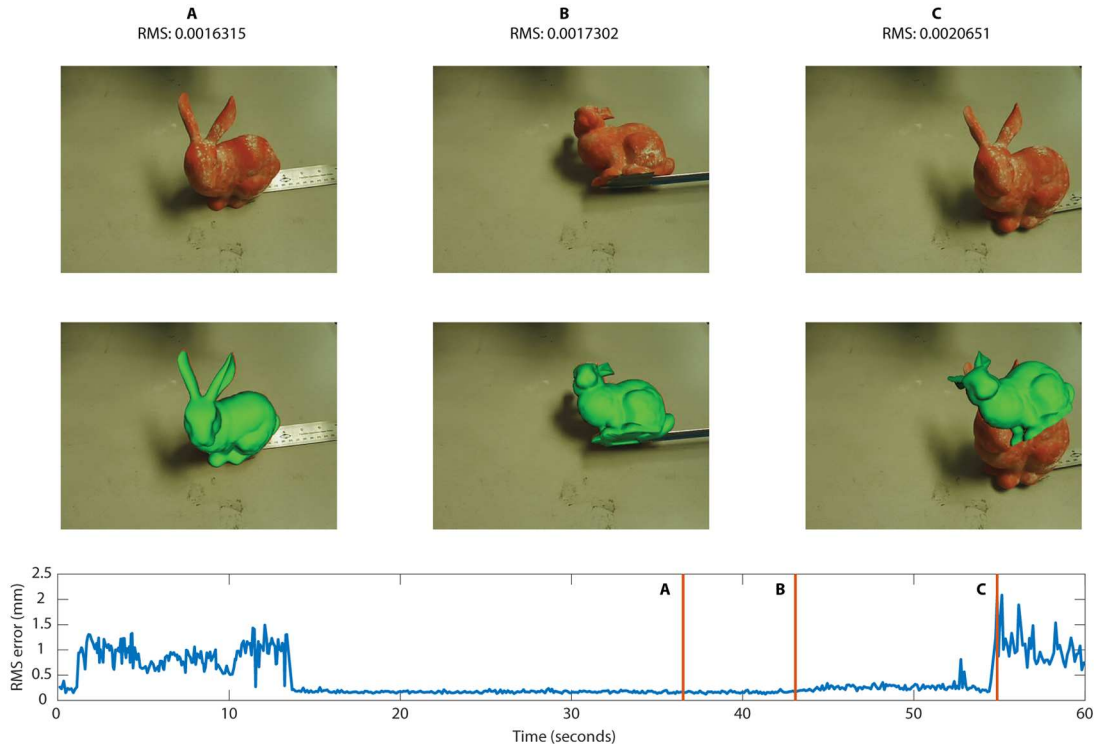
A
RMS: 0.0016315

B
RMS: 0.0017302

C
RMS: 0.0020651

Fig. 14. Top row: The bunny at three time instances A, B, and C as seen by the left stereo camera. Middle row: The estimated pose of the bunny, in green, superimposed on the stereo image. Bottom: RMS error versus time, for pose estimation using a Bingham filter, in an experiment that involved a user moving the bunny in the view of the camera.

Although the BF is able to handle moving objects, the image segmentation that we had previously been using to mask the point cloud received from the stereo cameras limited us to tracking static objects. We had been using simple a graph cut algorithm to segment our image, which required user input and could not be updated fast enough to keep up with the camera framerate. For this work, we augmented our segmentation by creating an automatic traveling mask. We begin by creating a mask using graph cut as before and using this mask for a rough registration. Once we are satisfied that the model is roughly registered to the object, we switch to using the traveling mask. Using the same rendering engine used in the graphical user interface (GUI), a z-depth buffer of the same size as the camera image is rendered of the camera's view of the model. This depth buffer is scaled from 0 to 255 with 0 representing the pixel farthest from the camera and 255 representing the closest. As empty pixels are read as infinitely far away they are limited to 0 in the depth buffer. Using this information, we create a new mask for our camera image by masking out all pixels with a depth of zero, effectively creating a cutout of our rendered model. Because we render the depth buffer every time the model's estimated transformation is changed, we create an image mask that moves along with our model.

We repeat the experiments with 3D printed pelvis and silicone prostate (see Figure 15). In the case of the pelvis, the RMS error in the first 2 s is around 1.5 mm. After this we apply a force on the pelvis in order to dislodge it from its location (see instance B in Figure 15). Upon application of an external force on the object, the pose estimation becomes erroneous ($\approx$ 2.7 mm). However, the BF is able to quickly recover after the movement and within 3 s the RMS error has been reduced to 1.8 mm (see instance C in Figure 15). Similarly, in the case of the prostate, the RMS error is initially around 1 mm (for example, instance D in Figure 15). We rotate and shift the prostate by about 2 inches in less than a second at around 17 s (instance E in Figure 15). The tracking RMS error increases to 2 mm, but it reduces to 0.8 mm in about 3 s.

Although the above experiments on object tracking show promising results, we have restricted ourselves to using repeated static pose updates. In theory, one could use a process model that better captures the motion of the object, which we leave for future work.

6. Conclusion and Discussions

In this work, a Bingham distribution-based linear filter (BF) was developed for online pose estimation. Bingham distribution captures the bimodal nature of the distribution of unit quaternions as well as the unit norm constraint. By adapting the linear measurement model developed by Srivatsan et al. (2016), a linear Bingham filter has been developed that updates the pose based on a pair of position measurements. Further the filter is extended to process
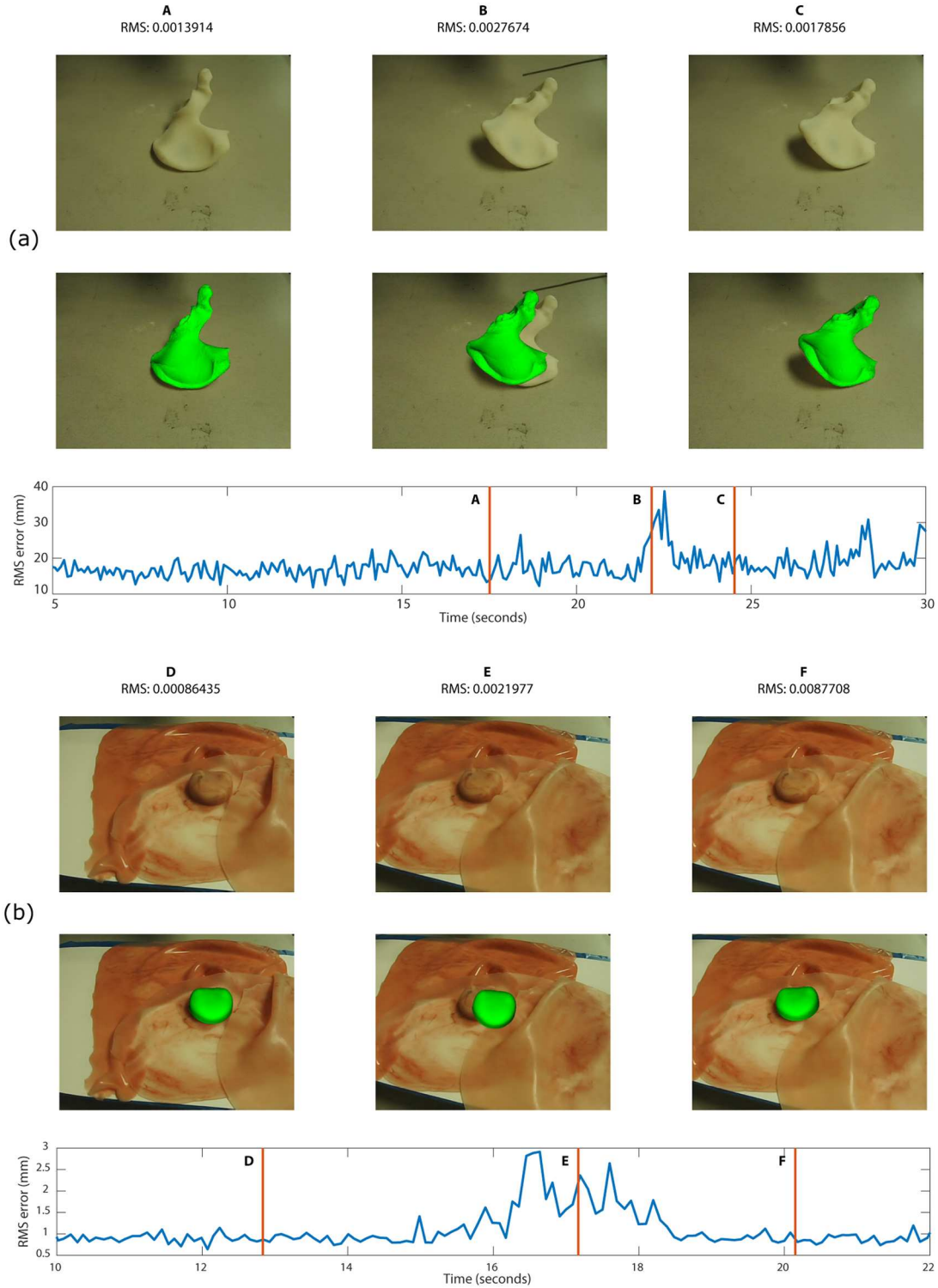
Fig. 15. (a) Stereo image-based tracking for a pelvis. Top row shows the left stereo camera image, middle row shows estimated pose superimposed, and the bottom plot shows the RMS error versus time. At instance B, the pelvis was poked by the user, which results in erroneous pose estimate. However, the pose estimate recovers to a low RMS error in a few seconds as shown in instance C. (b) Stereo image-based tracking for a prostate. Top row shows the left stereo camera image, middle row shows estimated pose superimposed and the bottom plot shows the RMS error versus time. At instance E, the prostate is shifted by about 2 inches in less than a second, and the Bingham filter loses track of the object. However, the pose estimate recovers to a low RMS error in a few seconds as shown in instance F.

surface-normal, pose as well as multiple simultaneous measurements. We demonstrate the efficacy and the versatility of our approach on a number of application in simulation and real-world experiments. The applications include sequential point registration, hand–eye calibration, object tracking, registration of stereo images, and point-cloud stitching.

It has been shown through simulations and experiments that the BF is capable of accurate pose estimation with less computation time compared with state-of-the-art methods. It is empirically observed that using multiple simultaneous measurements per update helps avoid local optima, when the correspondences are unknown. Further, we have developed a fast approach to data association that is based on creating a look-up table. This approach produces results that are several orders of magnitude faster than conventional k-d tree-based approaches.

One drawback of our approach, as with most filtering-based approaches, is that the estimate can be trapped in a local minima when the data association is unknown. Using a high initial uncertainty and more number of simultaneous measurements helps alleviate this problem to an extent. However, in some applications only pairs of measurements may be available per update, and the correspondences may be unknown (for example, probing-based registration). In such situations, better correspondences using a probabilistic metric as described by Billings et al. (2015), can improve the estimate. Another approach to resolve this issue is to use a global optimizer for filtering-based methods (Srivatsan and Choset, 2016).

Although the focus of this work was static pose estimation, we also demonstrated results for object tracking by running a series of static pose estimation. In the future, we plan to develop a process model to capture the dynamics of the moving object, and utilize an unscented Bingham filter (Gilitschenski et al., 2016) if this model is nonlinear. Another future direction involves using the estimate of the concentration matrix of the Bingham distribution to guide where to collect the next set of measurements that will improve the pose estimation.

ORCID iD

Rangaprasad Arun Srivatsan, ![ORCID] https://orcid.org/0000-0002-0863-5524

Notes

1. Here 10^{-300} is the smallest positive normalized floating-point number in IEEE® double precision.

2. The results are reported for a C++ implementation running on a MacBook Air, 1.6 GHz Intel Core i5 processor, 8 GB 1600 MHz DDR3 memory.

3. Source code available at: https://github.com/biorobotics/bingham_registration/tree/ ros-free

References

Ackerman MK, Cheng A, Shiffman B, Boctor E and Chirikjian G (2013) Sensor calibration with unknown correspondence: Solving $AX = XB$ using Euclidean-group invariants. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 1308–1313.

Besl PJ and McKay ND (1992) Method for registration of 3-D shapes. In: *Robotics-DL tentative*. International Society for Optics and Photonics, pp. 586–606.

Billings S and Taylor R (2014) Iterative most likely oriented point registration. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. New York: Springer, pp. 178–185.

Billings SD, Boctor EM and Taylor RH (2015) Iterative most-likely point registration (IMLP): A robust algorithm for computing optimal shape alignment. *PLoS one* 10(3): e0117688.

Bingham C (1974) An antipodally symmetric distribution on the sphere. *The Annals of Statistics* : 1201–1225.

Carlone L and Censi A (2014) From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization. *IEEE Transactions on Robotics* 30(2): 475–492.

Choukroun D, Bar-Itzhack I and Oshman Y (2006) Novel quaternion Kalman filter. *IEEE Transactions on Aerospace and Electronic Systems* 42(1): 174–190.

Daniilidis K (1999) Hand–eye calibration using dual quaternions. *The International Journal of Robotics Research* 18(3): 286–298.

Dornaika F and Horaud R (1998) Simultaneous robot-world and hand-eye calibration. *IEEE transactions on Robotics and Automation* 14(4): 617–622.

Engelhard N, Endres F, Hess J, Sturm J and Burgard W (2011) Real-time 3D visual SLAM with a hand-held RGB-D camera. In: *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, Vasteras, Sweden, vol. 180.

Estépar RSJ, Brun A and Westin CF (2004) Robust generalized total least squares iterative closest point registration. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. New York: Springer, pp. 234–241.

Faion F, Ruoff P, Zea A and Hanebeck U (2012) Recursive Bayesian calibration of depth sensors with non-overlapping views. In: *15th International Conference on Information Fusion*, pp. 757–762.

Fan T, Weng H and Murphey T (2017) Decentralized and recursive identification for cooperative manipulation of unknown rigid body with local measurements. *arXiv preprint arXiv:1709.01555*.

Faugeras OD and Hebert M (1986) The representation, recognition, and locating of 3-D objects. *The International Journal of Robotics Research* 5(3): 27–52.

Friedman JH, Bentley JL and Finkel RA (1977) An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software (TOMS) 3(3): 209–226.

Gilitschenski I, Kurz G, Julier SJ and Hanebeck UD (2014) A new probability distribution for simultaneous representation of uncertain position and orientation. In: *17th International Conference on Information Fusion*. IEEE, pp. 1–7.

Gilitschenski I, Kurz G, Julier SJ and Hanebeck UD (2016) Unscented orientation estimation based on the Bingham distribution. *IEEE Transactions on Automatic Control* 61(1): 172–177.

Glover J, Bradski G and Rusu RB (2012) Monte Carlo pose estimation with quaternion kernels and the distribution. *Robotics: Science and Systems* 7: 97.

Glover J and Kaelbling LP (2013) Tracking 3-D rotations with the quaternion Bingham filter. Technical Report MIT-CSAIL-TR-2013-005, MIT, Cambridge, MA.

Goddard JS and Abidi MA (1998) Pose and motion estimation using dual quaternion-based extended Kalman filtering. In: *Photonics West'98 Electronic Imaging*. International Society for Optics and Photonics, pp. 189–200.

Haines TS and Wilson RC (2008) Belief propagation with directional statistics for solving the shape-from-shading problem. In: *European Conference on Computer Vision*. New York: Springer, pp. 780–791.

Hauberg S, Lauze F and Pedersen KS (2013) Unscented Kalman filtering on Riemannian manifolds. *Journal of Mathematical Imaging and Vision* 46(1): 103–120.

Hillen T, Painter KJ, Swan AC and Murtha AD (2017) Moments of von mises and fisher distributions and applications. Mathematical Biosciences and Engineering: MBE 14(3): 673–694.

Horaud R and Dornaika F (1995) Hand-eye calibration. *The International Journal of Robotics Research* 14(3): 195–210.

Horn BK (1987) Closed-form solution of absolute orientation using unit quaternions. *JOSA A* 4(4): 629–642.

Izatt G, Dai H and Tedrake R (2017) Globally optimal object pose estimation in point clouds with mixed-integer programming. In: *International Symposium on Robotics Research*.

Jazwinski AH (2007) *Stochastic processes and filtering theory*. Courier Corporation.

Julier SJ and LaViola JJ (2007) On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing* 55(6): 2774–2784.

Kalman RE (1960) A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering* 82(1): 35–45.

Khoshelham K and Elberink SO (2012) Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors* 12(2): 1437–1454.

Kunze K and Schaeben H (2004) The Bingham distribution of quaternions and its spherical radon transform in texture analysis. *Mathematical Geology* 36(8): 917–943.

Kurz G, Gilitschenski I, Julier S and Hanebeck UD (2013) Recursive estimation of orientation based on the Bingham distribution. In: *16th International Conference on Information Fusion*, pp. 1487–1494.

LaViola Jr JJ (2003) A comparison of unscented and extended Kalman filtering for estimating quaternion motion. In: *Proceedings of the 2003 American Control Conference*, vol. 3. IEEE, pp. 2435–2440.

Lefebvre T, Bruyninckx H and De Schuller J (2002) Comment on "A new method for the nonlinear transformation of means and covariances in filters and estimators" [with authors' reply]. *IEEE Transactions on Automatic Control* 47(8): 1406–1409.

Marins JL, Yun X, Bachmann ER, McGhee RB and Zyda MJ (2001) An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4. IEEE, pp. 2003–2011.

McCartin-Lim M, McGregor A and Wang R (2012) Approximate principal direction trees. *arXiv preprint* arXiv:1206.4668.

Moghari MH and Abolmaesumi P (2007) Point-based rigid-body registration using an unscented Kalman filter. *IEEE Transactions on Medical Imaging* 26(12): 1708–1728.

Münch D, Combès B and Prima S (2010) A modified ICP algorithm for normal-guided surface registration. In: *Medical Imaging 2010: Image Processing*, vol. 7623. International Society for Optics and Photonics, 76231A.

Omohundro SM (1989) *Five balltree construction algorithms*. International Computer Science Institute Berkeley.

Onstott TC (1980) Application of the Bingham distribution function in paleomagnetic studies. *Journal of Geophysical Research: Solid Earth* 85(B3): 1500–1510.

Park FC and Martin BJ (1994) Robot sensor calibration: solving AX = XB on the Euclidean group, vol. 10, no. 5. IEEE Transactions on Robotics and Automation (Institute of Electrical and Electronics Engineers); (United States).

Pennec X and Thirion JP (1997) A framework for uncertainty and validation of 3-D registration methods based on points and frames. *International Journal of Computer Vision* 25(3): 203–229.

Pulli K (1999) Multiview registration for large data sets. In: *Proceedings Second International Conference on 3-D Digital Imaging and Modeling*. IEEE, pp. 160–168.

Richards P (1995) Constrained Kalman filtering using pseudomeasurements. In: *IEEE Colloquium on Algorithms for Target Tracking*. pp. 75–79.

Rosen DM, Carlone L, Bandeira AS and Leonard JJ (2016) SE-SYNC: A certifiably correct algorithm for synchronization over the special Euclidean group. *arXiv preprint arXiv:1612.07386*.

Rusinkiewicz S and Levoy M (2001) Efficient variants of the ICP algorithm. In: *Proceedings of 3rd International Conference on 3-D Digital Imaging and Modeling*. IEEE, pp. 145–152.

Segal A, Haehnel D and Thrun S (2009) Generalized-ICP. In: *Robotics: Science and Systems*, vol. 2.

Srivatsan RA, Ayvali E, Wang L, Roy R, Simaan N and Choset H (2016a) Complementary model update: A method for simultaneous registration and stiffness mapping in flexible environments. In: *IEEE International Conference on Robotics and Automation*. Stockholm, Sweden, pp. 924–930.

Srivatsan RA and Choset H (2016) Multiple start branch and prune filtering algorithm for nonconvex optimization. In: *The 12th International Workshop on The Algorithmic Foundations of Robotics*. New York: Springer.

Srivatsan RA, Rosen GT, Naina FD and Choset H (2016) Estimating SE(3) elements using a dual quaternion based linear Kalman filter. In: *Robotics: Science and Systems*.

Srivatsan RA, Xu M, Zevallos N and Choset H (2017) Bingham distribution-based linear filter for online pose estimation. In: *Proceedings of Robotics: Science and Systems*, Cambridge, MA.

Steinbring J and Hanebeck UD (2013) S 2 KF: The smart sampling Kalman filter. In: *2013 16th International Conference on Information Fusion (FUSION)*. IEEE, pp. 2089–2096.

Sturm J, Engelhard N, Endres F, Burgard W and Cremers D (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: *Proceedings of the International Conference on Intelligent Robot Systems*.

Tabb A and Yousef KMA (2015) Parameterizations for reducing camera reprojection error for robot-world hand-eye calibration. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 3030–3037.

Tsai RY and Lenz RK (1989) A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation* 5(3): 345–358.

Turk G and Levoy M (2005) The Stanford 3D Scanning Repository. Stanford University Computer Graphics Laboratory. Available at: http://graphics.stanford.edu/data/3Dscanrep.

Verma N, Kpotufe S and Dasgupta S (2009) Which spatial partition trees are adaptive to intrinsic dimension? In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, pp. 565–574.

Walker MW, Shao L and Volz RA (1991) Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding* 54(3): 358–367.

Yang J, Li H and Jia Y (2013) Go-ICP: Solving 3D registration efficiently and globally optimally. In: *IEEE International Conference on Computer Vision (ICCV)*. IEEE, pp. 1457–1464.

Yianilos PN (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. *SODA* 93: 311–321.

Zhou QY, Park J and Koltun V (2016) Fast global registration. In: *European Conference on Computer Vision*. New York: Springer, pp. 766–782.

Zhuang H, Roth ZS and Sudhakar R (1994) Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX=YB. *IEEE Transactions on Robotics and Automation* 10(4): 549–554.

Appendix A. Covariance of $h = G(x)b + c$

Proposition 1. *Let us consider $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ that are samples obtained from Gaussian distributions, $\mathcal{N}(0, \Sigma^b)$ and $\mathcal{N}(0, \Sigma^c)$, respectively. Let $x \in \mathbb{R}^l$ be obtained from a Gaussian distribution, $\mathcal{N}(\mu_x, \Sigma^x)$. Let $h \in \mathbb{R}^n$, and a linear matrix function $G(\cdot): \mathbb{R}^l \to \mathbb{R}^{n \times m}$, such that $h = G(x)b + c$. Assume that x, b and c are independent. Then Σ^h is given by*

$$\Sigma^h = G(\mu_x)\Sigma^b G^T(\mu_x) + N(\Sigma^b \circledast \Sigma^x)N^T + \Sigma^c \quad (38)$$

where $\circledast$ is the Kronecker product, $\Sigma^{\{\cdot\}}$ is the uncertainty associated with $\{\cdot\}$, and $N \in \mathbb{R}^{n \times lm}$ is defined as follows

$$N \triangleq [G_1 \ G_2 \ \cdots \ G_m]$$

Here $G_i \in \mathbb{R}^{n \times m}$ is obtained from the following identity,

$$G_i x = G(x)e_i$$

where e_i is the unit vector in $\mathbb{R}^m$ with 1 at position i and 0 everywhere else.

Example 6.1. Let us apply Proposition 1 to find uncertainty associated with $h = G(x)b + c$, where $x = (x_1, x_2)^T \in \mathbb{R}^2$.

Let us assume,

$$\Sigma^b = \begin{bmatrix} 0.7 & 0.01 \\ 0.01 & 4 \end{bmatrix}$$

$$\Sigma^c = \begin{bmatrix} 0.7 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}$$

$$G(x) = \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix}$$

$$\mu_x = (1, 0)^T \quad \text{and} \quad \Sigma^x = \begin{bmatrix} 1 & 0 \\ 0 & 0.1429 \end{bmatrix}$$

From Proposition 1, we obtain

$$\Sigma^h = \begin{bmatrix} 1.4723 & 0.1100 \\ 0.1100 & 4.2773 \end{bmatrix} \quad (39)$$

To verify the accuracy of this covariance, we perform a Monte Carlo experiment. We generate 100,000 samples of x, b, and c from their respective distributions. Here h is evaluated for each of the samples. We then find the covariance of the resulting samples of h. The calculated covariance is

$$\Sigma^h_{MC} = \begin{bmatrix} 1.4741 & 0.1081 \\ 0.1081 & 4.2757 \end{bmatrix} \quad (40)$$

Note that Σ^h_{MC} is similar to Σ^h.

Example 6.2. Let us apply Proposition 1 to find uncertainty associated with $h = G(x)b + c$, where $x = (x_1, x_2)^T \in \mathbb{R}^2$, and $|x|=1$. Let us assume, the same values for all parameters, except Σ^x. As x is constrained to have a unit norm, we obtain x from a Bingham distribution, $\mathcal{B}(M, Z)$ (instead of a Gaussian as in the previous example). As the Bingham distribution is a Gaussian distribution with a unit norm constraint as shown in Section 3.2.5 and Figure 3, we demonstrate with an example that Proposition 1 is valid even when x is obtained from a Bingham distribution instead of a Gaussian.

Let $M = I_{2 \times 2}$ and $Z = diag(0, -30)$. We first perform a Monte Carlo experiment with 100,000 of x, b and c obtained from their respective distributions. Here h is evaluated at each of these samples and are plotted as shown in Figure 16. Note that the points are distributed in the form of a Gaussian. The mean of the points is $(0.005, 0.016)^T$ and covariance is

$$\Sigma^h_{MC} = \begin{bmatrix} 1.4551 & 0.1263 \\ 0.1263 & 4.1624 \end{bmatrix} \quad (41)$$

From Equation (7), $\Sigma^x = -0.5 (M(Z + \lambda I)M^T)^{-1}$. Depending on the value of λ chosen, the value of Σ^x changes. We observe that choosing $\lambda = \min(z_i)$ results in a conservative estimate for the covariance. The estimated covariance is

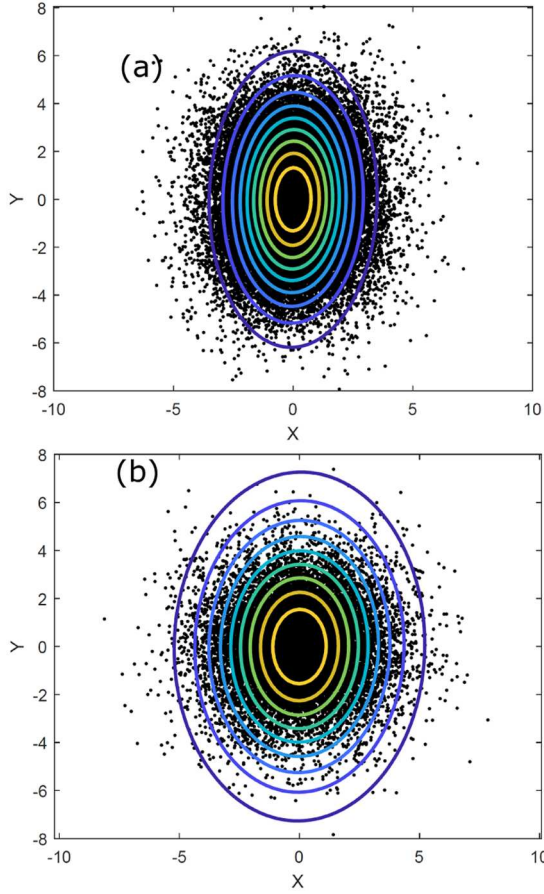$$\Sigma^h = \begin{bmatrix} 1.4450 & 0.1101 \\ 0.1101 & 4.2725 \end{bmatrix} \quad (42)$$

Fig. 16. PDF contours obtained from the covariance estimated using Proposition 1: (a) $Z = diag(0, -30)$; (b) $Z = diag(0, -3)$. The black points are samples of $h = G(x)b + c$ obtained from 100,000 Monte Carlo simulations.

Here Σ_{MC}^h and Σ^h are in good agreement with each other. Figure 16(a) shows PDF contours for the estimated Σ^h and the samples from the Monte Carlo simulation. We repeat this experiment for $Z = diag(0, -3)$. The PDF contours as well as 100,000 points obtained from Monte Carlo simulation as shown in Figure 16(b). Note how the estimated PDF accurately describes distribution of the point samples obtained from Monte Carlo simulation.

Appendix B. Estimating the likelihood of obtaining t

From Equation (15),

$$\widetilde{t} = \frac{(\widetilde{a}_1 + \widetilde{a}_2) - \widetilde{q} \odot (\widetilde{b}_1 - \widetilde{b}_2) \odot \widetilde{q}^*}{2}$$
$$= \widetilde{a}_c - \widetilde{q} \odot \widetilde{b}_c \odot \widetilde{q}^*$$

where $\widetilde{d}_c = \frac{\widetilde{d}_1 + \widetilde{d}_2}{2}$, $d = a, b$,

$$\Rightarrow \widetilde{t} \odot \widetilde{q} = \widetilde{a}_c \odot \widetilde{q} - \widetilde{q} \odot \widetilde{b}_c$$
$$\Rightarrow F_2(\widetilde{q})\widetilde{t} = F_2(\widetilde{q})\widetilde{a}_c - F_1(\widetilde{q})\widetilde{b}_c \quad \text{from Equation (1)}$$
$$\Rightarrow W_1 t = W_1 a_c - W_2 b_c \tag{43}$$

where $W_1, W_2 \in \mathbb{R}^{4 \times 3}$ are the last three columns of $F_2(\widetilde{q})$ and $F_1(\widetilde{q})$, respectively.

From Section 4.1.1, $a_i^s = a_i + \delta a_i$ and $b_i^s = b_i + \delta b_i$. Substituting these terms in Equation (43) gives

$$W_1 t - W_1 a_c^s + W_2 b_c^s + G_2(\widetilde{q})\mu = 0$$

where $\mu = (\delta a_1, \delta a_2, \delta b_1, \delta b_2)^T$ and

$$G_2 = \frac{1}{2}\begin{bmatrix} W_1 & W_1 & -W_2 & -W_2 \end{bmatrix}$$

As observed earlier in the case of estimation of $\widetilde{q}$, $G_2(\widetilde{q})\mu$ is a zero mean Gaussian noise, $\mathcal{N}(0, R)$, where the uncertainty R is obtained analytically similar to the evaluation of Q from Proposition 1.