Article

The International Journal of Robotics Research
1–17

DOI: 10.1177/0278364919842324
journals.sagepub.com/home/ijr
SAGE

Registration with a small number of sparse measurements

Rangaprasad Arun Srivatsan, Nicolas Zevallos, Prasad Vagdargi and Howie Choset

Abstract

This work introduces a method for performing robust registration given the geometric model of an object and a small number (less than 20) of sparse point and surface normal measurements of the object's surface. Such a method is of critical importance in applications such as probing-based surgical registration, contact-based localization, manipulating objects devoid of visual features, etc. Our approach for sparse point and normal registration *(SPNR) is iterative in nature. In each iteration, the current best pose estimate is perturbed to generate several candidate poses. Among the generated poses, one pose is selected as the best, by evaluating an inexpensive cost function. This pose is used as the initial condition to estimate the locally optimum registration. This process is repeated until the registration estimate converges within a tolerance bound. Two variants are developed: deterministic (dSPNR) and probabilistic (pSPNR). The dSPNR is faster than pSPNR in converging to the local optimum, but the pSPNR requires fewer parameters to be tuned. The pSPNR also provides pose-uncertainty information in addition to the registration estimate. Both approaches were evaluated in simulation using various standard datasets and then compared with results obtained using state-of-the-art methods. Upon comparison with other methods, both dSPNR and pSPNR were found to be robust to initial pose errors as well as noise in measurements. The effectiveness of the approaches are also demonstrated with robot experiments for the application of probing-based registration.*

Keywords

Kalman filter, Bayes rule, registration

1. Introduction

In several applications of engineering, medicine, and especially robotics, one often encounters the need to perform registration. In a typical registration problem, the spatial transformation between the geometric model of the object-of-interest and point/normal measurements of the object's surface needs to be estimated. In most applications, the point clouds obtained from sensors such as Lidar, Kinect, stereo cameras, etc., contain hundreds of points. Several methods have been developed to perform registration when dense point measurements are obtained (Audette et al., 2000; Besl and McKay, 1992; Billings et al., 2015; Moghari and Abolmaesumi, 2007; Rusinkiewicz and Levoy, 2001; Segal et al., 2009). There also exist registration methods that deal with surface normal measurements, in addition to point measurements (Billings and Taylor, 2014; Münch et al., 2010; Pulli, 1999). However, these methods do not perform well when only a small number of sparsely distributed measurements are available (as observed by Glozman et al., 2001), and hence in this work we develop a method for robust sparse point and normal registration (SPNR).

Registration with sparse measurements is of critical importance in surgical applications, where a surgeon probes the visible anatomy using a robot in order to register the anatomy to its preoperative model obtained from computed tomography (CT) scan or magnetic resonance imaging (MRI) (Ma and Ellis, 2003; Simon et al., 1995). In such applications, there is a cost associated with probing more locations and the goal is to quickly and accurately register with a small number of measurements (see Figure 1). Prior work either uses a priori knowledge of anatomical landmarks to hand-pick a small number of probing locations (Ma and Ellis, 2003, 2004b; Simon et al., 1995), or perform

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

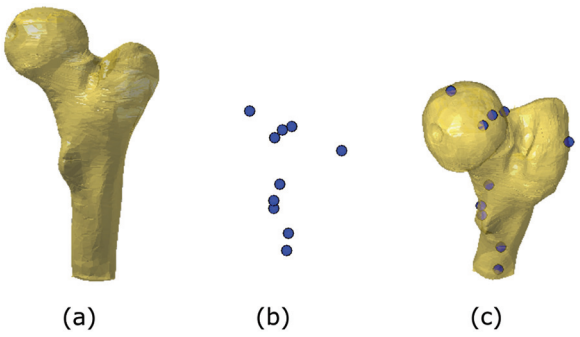

Fig. 1. (a) Geometric model of the object. (b) Point measurements in the sensor frame. (c) Point measurements registered to the geometric model.

a pre-registration by asking an expert to manually find point correspondences. In an attempt to keep the formulation general, in this work we do not assume any prior knowledge of anatomical segments.

Our approach for SPNR is developed as an iterative procedure and, in each iteration, the current *best pose* estimate is perturbed to obtain several poses. Each of the obtained poses will hereby be referred to as a "pose particle." The amount of perturbation is reduced in each iteration to balance exploration and exploitation. By evaluating a cost function, the best pose particle is selected and used as the initial seed for an optimization problem that computes a locally optimal pose. This process is then repeated for a fixed number of iterations or until convergence.

The optimizer used for computing the locally optimal pose can be deterministic or probabilistic and depending on the requirement of the problem, two variants have been developed: deterministic SPNR (dSPNR) and probabilistic SPNR (pSPNR). The dSPNR uses iterative most likely oriented point (IMLOP) algorithm (Billings and Taylor, 2014), whereas the pSPNR uses dual quaternion filtering (DQF) (Srivatsan et al., 2016b) to estimate the pose.

The dSPNR is computationally faster than pSPNR, but it requires the perturbation related parameters to be set manually, unlike the pSPNR, that uses uncertainty information to automatically set these parameters. An additional contribution of this work is extending the online formulation of DQF in our prior work (Srivatsan et al., 2016b) to process batches of point measurements as well as surface-normal measurements.

This paper is an improved and extended version of our prior work (Srivatsan et al., 2017a). The improvements include an extension of our prior approach to use both point and normal measurements when available. We provide a more detailed study on how sensitive our algorithms are to the tuning of various parameters. Finally, while our prior work looks at registration using automated probing, in this work we include an additional robot experiment where a surgical robot is telemanipulated to probe an object at various locations and the measurements are used to perform registration.

The paper is organized as follows. The related work in the literature is discussed in Section 2. We present the mathematical background required to formulate our approach for SPNR in Section 3. In Section 4, we describe the formulation of dSPNR and pSPNR and discuss the various parameters that need to be tuned. Following that, the dSPNR and pSPNR are evaluated in simulation over a number of standard datasets and compared against popular registration methods in Section 5. The results show that the SPNR typically takes less than 20 point measurements to accurately and robustly estimate the registration and is more robust to initial registration errors ($\approx 30^\circ$ orientation and ≈ 30 mm translation). When considering surface normal measurements alongside point measurements, the estimated registration is more accurate than using point measurements alone, when using only 10–15 measurements. In Section 6, we describe robot experiments for automated as well as telemanipulated probing-based registration of an object of interest. A general guideline is also provided on how to automatically probe the object to get a good spread of sparse points.

2. Related work

Registration is the process of finding the spatial transformation that aligns a point cloud to a geometric model, defined in different reference frames. When the correspondences between the points in the two reference frames are known, Faugeras and Hebert (1986), Horn (1987), Arun et al. (1987), andWalker et al. (1991) have developed deterministic approaches to find the spatial transformation. In our prior work (Srivatsan et al., 2018), we introduced a filtering-based probabilistic approach to find the globally optimal registration when the correspondences are known. However, in most practical applications, the correspondence is unknown. When the correspondence is unknown, the most popular approach involves iteratively finding the best correspondence and then the optimal transformation given that correspondence. Such an approach was popularized by Besl and McKay (1992), who developed the iterative closest point (ICP) algorithm.

Several variants of the ICP have since been developed that use surface normal information (Billings and Taylor, 2014; Rusinkiewicz and Levoy, 2001; Tam et al., 2013), incorporate uncertainty in measurements (Estéepar et al., 2004; Segal et al., 2009), incorporate uncertainty in finding correspondence (Billings et al., 2015; Granger and Pennec, 2002), are robust to outliers (Phillips et al., 2007; Tsin and Kanade, 2004), are globally optimal (Gelfand et al., 2005; Izatt et al., 2017; Makadia et al., 2006; Yang et al., 2013), etc.

The computation time for most of the ICP-based methods is high and in order to address this, Kalman filtering-based variants have been developed (Hauberg et al., 2013; Moghari and Abolmaesumi, 2007; Pennec and Thirion, 1997; Srivatsan and Choset, 2016). There has also been

work on particle filtering-based methods (Glover et al., 2012; Ma and Ellis, 2004b) and Bingham filtering-based methods for registration (Srivatsan et al., 2018).

In applications where only a small number of sparse measurements, conventional registration approaches do not perform reliably (Glozman et al., 2001). For example, in a surgical application of probing-based registration using any of the Kalman filtering variants, the computation time might be low (Moghari and Abolmaesumi, 2007; Srivatsan et al., 2016a,b), but the time taken to obtain greater than 100 point measurements can be high.

When the correspondence between the measurements and the geometric model is known, registration is non-convex, but the optimal parameters can still be estimated using the approaches of Horn (1987), Arun et al. (1987), Faugeras and Hebert (1986), Walker et al. (1991), and Srivatsan et al. (2018). But when the number of measurements available is small with unknown correspondence, registration becomes highly non-convex and any optimization technique used has the potential to become trapped in local minima. Two approaches, prevalent in literature, counteract the issue of local minima; these are (1) using global optimizers and (2) finding correspondences using heuristics or additional information.

The Go-ICP of Yang et al. (2013) is an example of an approach that uses a branch and bound global optimizer to find the registration. However, the computational time is large and the method does not extend to using surface normal measurements. Other examples of global optimizers for registration include using genetic algorithms (Seixas et al., 2008), simulated annealing (Luck et al., 2000), and particle swarm optimization (Wachowiak et al., 2004). Further, there exist methods that use convex relaxation to repose registration as a convex optimization problem (Horowitz et al., 2014; Izatt et al., 2017; Maron et al., 2016). However, these approaches are computationally expensive and restrictive in that they cannot be trivially extended to incorporate sensor noise, pose uncertainty, or surface normal measurements.

Examples of methods that use heuristics or additional information to find correspondence include the works of Simon et al. (1995) and Ma and Ellis (2003, 2004a). In such methods, registration is performed by probing at carefully selected locations. These points are selected based on optimization of a stiffness-based quality metric (Ma and Ellis, 2004a). Since these works were inspired by surgical applications, they assume a priori knowledge of the location of anatomical features in the robot frame. Such an assumption is very limiting in its nature and reduces the scope of the approach. First, the workspace constraints of the robot might restrict the robot from locating all the anatomical segments. Second, when applied to a non-surgical domain, defining an equivalent of anatomical feature is non-trivial. The approach of Glozman et al. (2001) avoids choosing probing locations based on the stiffness metric, but instead precomputes a small number of best locations to probe by perturbing the model and computing which subset of points are most sensitive to registration error when selected. Following this, an expert surgeon manually chooses these points on the actual anatomy and a pre-registration step is performed. This step helps reduce the initial registration error greatly and eventually leads to an accurate registration. Our approach is influenced by Glozman et al. (2001) in that we use geometric mesh models of low resolution for the initial computations and use the finest level of resolution for the final computations. Our approach, however, does not require an expert to perform pre-registration.

The approach of Gelfand et al. (2005); Rusu et al. (2009), Jiang et al. (2009), and Makadia et al. (2006) calculate features in the point cloud and use the calculated features to estimate the registration. A major drawback of these approaches is that the feature computation requires dense point cloud, and hence are unsuitable when we have sparse measurements. Additional information such as color from camera images can be used to perform robust registration, as demonstrated by Xiang et al. (2017), Mitash et al. (2018), and Li et al. (2018).

There also exist methods that probabilistically assign correspondence when performing registration. These methods are also referred to as softassign algorithms (Agamennoni et al., 2016; Eckart et al., 2015; Gold et al., 1998; Jian and Vemuri, 2005; Myronenko and Song, 2010; Tsin and Kanade, 2004). These methods typically represent the point cloud using a Gaussian mixture model (GMM), and derive an optimization framework for registration, that they claim to be robust, although at the cost of higher computational complexity. The most popular among this class of methods is coherent point drift (CPD) developed by Myronenko and Song (2010). Recently, Eckart et al. (2015) developed an approach that improves the computational complexity of CPD by introducing latent variables. Although these methods are also only locally optimal similar to ICP, they are more robust to sensor noise and have a wider basin of convergence.

There is also a vast literature on touch-based localization for robot manipulation applications. These approaches are either meant for planar localization (Chhatpar and Branicky, 2005; Gadeyne and Bruyninckx, 2001; Hsiao and Kaelbling, 2010), are time consuming (Petrovskaya and Khatib, 2011), or assume good initial alignment (Saund et al., 2017) or assume multiple simultaneous contacts (Hebert et al., 2013; Javdani et al., 2013).

The work of Ma and Ellis (2004b) and stochastic-ICP of Penney et al. (2001) come closest to our approach. Ma and Ellis (2004b) used an unscented particle filter (UPF) to register an object using a small number of point measurements without relying on prior knowledge of any landmarks or segments. Although the UPF uses a small number of measurements, it uses a large number of pose particles ($\approx 2,000$) in each iteration, resulting in a large computation time. Stochastic-ICP of Penney et al. (2001) is similar to the ICP, with the exception that in each iteration, every measurement is perturbed by adding a random noise. This is different from our approach where we perturb the pose

Table 1. Femur bone: registration in the presence of noise.

	No noise		2 mm noise		5 mm noise		10 mm noise	
20 points	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)
dSPNR	0.08	0	0.44	1.08	0.46	1.79	0.56	7.03
pSPNR	0.07	0	1.00	1.63	0.99	1.66	1.26	6.94
ICP	0.01	4.72	0.01	2.92	0.01	6.69	0.01	7.40
CPD	0.87	6.27	0.81	6.17	0.75	6.69	0.94	10.29
bDQF	0.05	2.84	0.09	2.19	0.09	7.83	0.1	13.64
UPF	161.77	7.13	159.83	22.12	242.38	24.23	248.56	32.34
GoICP	20.29	2.16	22.25	1.65	21.71	4.51	21.81	7.22
SparseICP	0.05	41.18	0.06	41.49	0.04	26.91	0.04	24.56
100 points								
dSPNR	0.03	0	0.51	0.59	0.53	1.12	0.82	1.64
pSPNR	0.13	0	2.53	0.64	3.10	1.07	2.88	1.48
ICP	0.01	0	0.01	1.39	0.01	1.93	0.01	1.64
CPD	1.29	1.41	1.35	1.75	1.39	1.99	1.62	3.99
bDQF	0.08	0	0.19	0.38	0.24	1.55	0.24	2.25
UPF	1891.1	4.82	1843.3	8.06	1594.0	14.29	1621.5	17.91
GoICP	20.28	0.00	20.06	0.86	20.18	1.72	39.75	1.64
SparseICP	0.15	21.20	0.11	26.98	0.11	28.80	0.11	35.29

parameters in each iteration which results in each measurement being perturbed in a structured manner.

It is also worth mentioning that there a number of approaches in the literature that deal with registration of incomplete point clouds (also referred to as *sparse* point clouds). Some examples include SparseICP of Bouaziz et al. (2013), Luong et al. (2016), and Yuan et al. (2018), who dealt with the registration of an incomplete point cloud obtained from sensors such as lidars. Bouaziz et al. (2013) reformulated the objective function of ICP with a sparsity inducing norm. Luong et al. (2016) used a probabilistic data association similar to CPD. Yuan et al. (2018) used an encoder–decoder network to complete the point cloud, using a network trained on similar point cloud datasets. An ICP is then used to perform registration with the generated dense point cloud. A major difference between these methods and ours is that we deal with sparse and small numbers of measurements. The small number of measurements poses a challenge to these existing algorithms and results in poor estimates (see Table 1).

3. Mathematical background

3.1. ICP

The ICP algorithm introduced by Besl and McKay (1992) has two important steps:

1. estimate the correspondences between the point cloud and the model using a closest point rule, given the current best estimate of registration;
2. compute the transformation which minimizes the distance between corresponding points.

These two steps are repeated until convergence. The convergence criteria is usually a maximum number of iterations or the change in rotation and translation falling below a user set threshold.

Consider two point clouds, $A = \{\boldsymbol{a}_i\}$, $\boldsymbol{a}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$, are n points on the geometric model of the object and $B = \{\boldsymbol{b}_j\}$, $\boldsymbol{b}_j \in \mathbb{R}^3$, $j = 1, \ldots, m$, are m points obtained using sensor measurements. Let $\boldsymbol{T} \in SE(3)$ be the transformation that aligns A and B. The point $\boldsymbol{c}_j$ is the current best estimate for the model point that corresponds to sensor point $\boldsymbol{b}_j$, and is obtained using a closest point rule:

$$\boldsymbol{c}_j = \arg\min_{\boldsymbol{c} \in A} \left\| \boldsymbol{c} - \boldsymbol{T}(\boldsymbol{b}_j) \right\|^2$$

where $\boldsymbol{T}(\boldsymbol{b}_j) = \boldsymbol{R}\boldsymbol{b}_j + \boldsymbol{t}$, $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is the rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector. The correspondence rule described above is referred to as "FindClosestPoint" in Algorithm 1. The ICP algorithm listed in Algorithm 1, typically uses the approach of Horn (1987) to perform the minimization; although there are other optimization variants as well, such as the approach of Tam et al. (2013).

The complexity of original implementation of ICP is $O(nm)$. However, using a kd-tree search to estimate correspondence reduces the complexity to $O(m \log(n))$.

3.2. IMLOP

In some applications, in addition to position measurements, surface-normal measurements may also be available (Billings et al., 2015; Glozman et al., 2001; Pulli, 1999; Srivatsan et al., 2016a). Note that we cannot obtain registration estimates using only surface normal measurements. We need position and normals to estimate registration.

Algorithm 1. ICP Update

Input:
$A = \{\boldsymbol{a}_i \in \mathbb{R}^3\},\ i = 1, 2, \ldots, n$
$B = \{\boldsymbol{b}_j \in \mathbb{R}^3\},\ j = 1, 2, \ldots, m$
Initial transformation: $\boldsymbol{T}_0 \in SE(3)$
Output: $\boldsymbol{T} \in SE(3)$ that aligns A and B
Initialize: $\boldsymbol{T} \leftarrow \boldsymbol{T}_0$
while *not converged* **do**
  Correspondence: $\boldsymbol{c}_j = \text{FindClosestPoint}(\boldsymbol{T}(\boldsymbol{b}_j)),\ \boldsymbol{c}_j \in A$
  Minimization: $\boldsymbol{T} = \arg\min_{\boldsymbol{T}} \sum_{j=1}^{m} \left\|\boldsymbol{c}_j - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2$

Without position measurements, only orientation estimation is possible, not translation. It is common practice to assume that when normal measurements are available, a corresponding point measurement is also available (Billings and Taylor, 2014; Glozman et al., 2001).

The IMLOP algorithm, developed by Billings and Taylor (2014) is a variation of the ICP that considers both point and normal measurements. Instead of a closest point rule to find the correspondence, IMLOP uses a most likely oriented point rule,

$$(\boldsymbol{c}_j, \boldsymbol{n}^{\boldsymbol{c}_j}) = \arg\min_{(\boldsymbol{c}, \boldsymbol{n}^{\boldsymbol{c}}) \in A} \frac{1}{2\sigma^2} \left\|\boldsymbol{c} - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2 - \kappa (\boldsymbol{n}^{\boldsymbol{c}})^{\mathrm{T}} \boldsymbol{R} \boldsymbol{n}^{\boldsymbol{b}_j}$$

where σ^2 and κ are residual match errors. The expression for σ^2 is simply estimated as the mean square distance between the matches divided by the spatial dimensionality, $\sigma^2 = \frac{1}{3n}\sum_j \left\|\boldsymbol{c}_j - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2$. The residual error in surface normals is estimated using an approximation as derived by Banerjee et al. (2005), $\kappa = h(\boldsymbol{b}_j, \boldsymbol{c}_j, \boldsymbol{n}^{\boldsymbol{b}_j}, \boldsymbol{n}^{\boldsymbol{c}_j})$, where the expression of κ is provided in (5.4) of Billings (2015). We refer to this correspondence rule as "FindClosestOrientedPoint" in Algorithm 2.

Billings (2015) showed that the minimization can be solved in closed form using a modification of the singular value decomposition (SVD) method of Arun et al. (1987) (see Algorithm 5.3 in Billings (2015)). The algorithm for IMLOP is shown in Algorithm 2. When surface normal measurements are unavailable, setting $\kappa = 0$, simplifies IMLOP to ICP. The complexity of IMLOP is $O(m \log(n))$, similar to ICP.

Algorithm 2. IMLOP Update

Input:
$A = \{\boldsymbol{a}_i, \boldsymbol{n}^{\boldsymbol{a}_i} \in \mathbb{R}^3\},\ i = 1, 2, \ldots, n$
$B = \{\boldsymbol{b}_j, \boldsymbol{n}^{\boldsymbol{b}_j} \in \mathbb{R}^3\},\ j = 1, 2, \ldots, m$
Initial transformation: $\boldsymbol{T}_0 \in SE(3)$
Initial residual noise parameters: $\kappa_0,\ \sigma_0^2$
Output: $\boldsymbol{T} \in SE(3)$ that aligns A and B
Initialize: $\boldsymbol{T} \leftarrow \boldsymbol{T}_0,\ \kappa \leftarrow \kappa_0,\ \sigma^2 \leftarrow \sigma_0^2$
while *not converged* **do**
  Correspondence: $(\boldsymbol{c}_j, \boldsymbol{n}^{\boldsymbol{c}_j} \in A) =$ FindClosestOrientedPoint$(\boldsymbol{T}(\boldsymbol{b}_j), \boldsymbol{R}(\boldsymbol{n}^{\boldsymbol{b}_j}), \sigma^2, \kappa)$
  Minimization:
  $\boldsymbol{T} = \arg\min_{\boldsymbol{T}} \frac{1}{2\sigma^2} \sum_{j=1}^{m} \left\|\boldsymbol{c}_j - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2 - \kappa \sum_{j=1}^{m} (\boldsymbol{n}^{\boldsymbol{c}_j})^{\mathrm{T}} \boldsymbol{R} \boldsymbol{n}^{\boldsymbol{b}_j}$
  Update residual noise: $\sigma^2 = \frac{1}{3n}\sum_j \left\|\boldsymbol{c}_j - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2$
  $\kappa = (\boldsymbol{b}_j, \boldsymbol{c}_j, \boldsymbol{n}^{\boldsymbol{b}_j}, \boldsymbol{n}^{\boldsymbol{c}_j})$

3.3. DQF with point measurements

DQF is a linear Kalman filtering-based approach for online pose estimation (Srivatsan et al., 2016b). Unlike ICP, DQF is not a batch processing algorithm. It is an online processing algorithm, similar to a Kalman filter, as it uses measurement information as it becomes available, but with one small difference: the DQF updates the registration once for every pair of measurements obtained.

Compared with other filtering-based registration methods such as the works of Pennec and Thirion (1997), Moghari and Abolmaesumi (2007), and Hauberg et al. (2013), the DQF is preferred because it is a truly linear filter without any approximations or linearizations, resulting in quick and accurate estimates. The transformation $\boldsymbol{T}$ is parameterized using the unit dual quaternion $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{d})^{\mathrm{T}}$, where $\boldsymbol{q} \in \mathbb{R}^4$ is a unit quaternion that parameterizes the rotation, $\boldsymbol{d} = \left((0, \boldsymbol{t})^{\mathrm{T}} \odot \boldsymbol{q}\right)/2$, where $\odot$ is the quaternion multiplication operator.

In our prior work (Srivatsan et al., 2016b), we used a pair of point measurements $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j+1}$ in each iteration to update the registration estimate. Let $\boldsymbol{c}_j$ and $\boldsymbol{c}_{j+1}$ be the best estimate for the points on the model that correspond to $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j+1}$, respectively. As with ICP, the best estimate for the correspondence given the current estimate for registration, is obtained using a closest point rule.

The relation between the sensor points and corresponding model points, assuming perfect correspondence and no noise, is

$$\widetilde{\boldsymbol{c}}_j = \boldsymbol{q} \odot \widetilde{\boldsymbol{b}}_j \odot \boldsymbol{q}^* + \widetilde{\boldsymbol{t}} \tag{1}$$

$$\widetilde{\boldsymbol{c}}_{j+1} = \boldsymbol{q} \odot \widetilde{\boldsymbol{b}}_{j+1} \odot \boldsymbol{q}^* + \widetilde{\boldsymbol{t}} \tag{2}$$

where $\widetilde{\boldsymbol{y}} = [0, \boldsymbol{y}^{\mathrm{T}}]^{\mathrm{T}}, \forall\, \boldsymbol{y} \in \mathbb{R}^3$. By subtracting (2) from (1) and rearranging the terms, we obtain

$$\boldsymbol{H}\boldsymbol{q} = \boldsymbol{0}, \quad \boldsymbol{H} \in \mathbb{R}^{4\times 4} \tag{3}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & -(\boldsymbol{c}_j^{j+1} - \boldsymbol{b}_j^{j+1})^{\mathrm{T}} \\ (\boldsymbol{c}_j^{j+1} - \boldsymbol{b}_j^{j+1}) & (\boldsymbol{c}_j^{j+1} + \boldsymbol{b}_j^{j+1}) \times \end{bmatrix} \tag{4}$$

$$\boldsymbol{t} = \frac{\boldsymbol{c}_j^{j+1}}{2} - \text{Real}\left(\boldsymbol{q} \odot \frac{\boldsymbol{b}_j^{j+1}}{2} \odot \boldsymbol{q}^*\right) \tag{5}$$

where $\boldsymbol{c}_j^{j+1} = \boldsymbol{c}_j - \boldsymbol{c}_{j+1},\ \boldsymbol{b}_j^{j+1} = \boldsymbol{b}_j - \boldsymbol{b}_{j+1},\ j = 1, 3, \ldots, 2\alpha - 1,\ \alpha = \left\lfloor \frac{m}{2} \right\rfloor$, m is the total number of measurements,

$\boldsymbol{q}^*$ is the conjugate quaternion and $[\,]^\times$ is the operator that converts a vector to a skew-symmetric matrix and $\text{Real}(\boldsymbol{y}) = [y_2, y_3, y_4]^\text{T}$ for all $\boldsymbol{y} = [y_1, y_2, y_3, y_4]^\text{T} \in \mathbb{R}^4$. For more details on the derivation, refer to Srivatsan et al. (2016b).

Since noise exists in the point measurements and the correspondence is not perfect, solving for $\boldsymbol{q}$ from (4) does not directly give us the rotation. We instead use a Kalman filter to estimate $\boldsymbol{q}$ iteratively. The update equations for the Kalman filter are

$$\begin{aligned}\boldsymbol{q}_k &= \boldsymbol{q}_{k-1} - \boldsymbol{K}_k \boldsymbol{H}_k \boldsymbol{q}_{k-1} \\ \Sigma^{\boldsymbol{q}}_k &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\Sigma^{\boldsymbol{q}}_{k-1}\end{aligned}$$

where

$$\boldsymbol{K}_k = \Sigma^{\boldsymbol{q}}_{k-1} \boldsymbol{H}_k^\text{T} (\boldsymbol{H}_k\, \Sigma^{\boldsymbol{q}}_{k-1} \boldsymbol{H}_k^\text{T} + \boldsymbol{Q}_k)^{-1}$$

where $\Sigma^{\boldsymbol{q}}_k$ is the uncertainty in the quaternion $\boldsymbol{q}_k$ and $\boldsymbol{Q}_k$ is the uncertainty in $\boldsymbol{z}_k$, where $\boldsymbol{z}_k \triangleq \boldsymbol{H}_k q_{k-1}$. The translation vector $\boldsymbol{t}_k$ is obtained from $\boldsymbol{q}_k$ using (5). For the sake of brevity, the derivation for $\boldsymbol{Q}_k = \boldsymbol{g}(\Sigma^{\boldsymbol{q}}_k, \Sigma^{\boldsymbol{b}_j}_k, \Sigma^{\boldsymbol{c}_j}_k, \Sigma^{\boldsymbol{b}_{j+1}}_k, \Sigma^{\boldsymbol{c}_{j+1}}_k)$ and $\Sigma^{\boldsymbol{t}}_k = \boldsymbol{f}(\Sigma^{\boldsymbol{q}}_k, \Sigma^{\boldsymbol{b}_j}_k, \Sigma^{\boldsymbol{c}_j}_k, \Sigma^{\boldsymbol{b}_{j+1}}_k, \Sigma^{\boldsymbol{c}_{j+1}}_k)$ representing the uncertainty in the translation, are omitted here. The expressions along with their derivation can be obtained from Section IV.C of Srivatsan et al. (2016b).

The standard implementation of DQF requires ≈100 point measurements for reliable registration estimation (Srivatsan et al., 2016b). In the previous version of this paper (Srivatsan et al., 2017a), we modified the DQF to a batch processing variant, which updates using all the m measurements collected, instead of a single pair per iteration. We shall henceforth refer to this variant as batch DQF (bDQF). As shown in Section 5, bDQF requires fewer measurements for accurate estimation compared with the standard DQF. We modify (3) as

$$\boldsymbol{G}\boldsymbol{q} = 0, \quad \boldsymbol{G} \in \mathbb{R}^{\alpha \times 4}, \quad \text{where } \boldsymbol{G} = [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_\alpha] \tag{6}$$

Here $\boldsymbol{H}_i$ is as defined in (4). The update equations of the filter remain the same. The algorithm for bDQF is shown in Algorithm 3.

Algorithm 3: bDQF with point measurements

Input:
$A = \{\boldsymbol{a}_i \in \mathbb{R}^3\},\ i = 1, 2, \ldots, n$
$B = \{\boldsymbol{b}_j \in \mathbb{R}^3\},\ j = 1, 2, \ldots, m$
Initial transformation: $\boldsymbol{q}_0 \in \mathbb{R}^4, \boldsymbol{t}_0 \in \mathbb{R}^3$,
Output:
$\boldsymbol{q} \in \mathbb{R}^4, \boldsymbol{t} \in \mathbb{R}^3$ that aligns A and B
$\Sigma^q \in \mathbb{R}^{4\times 4}, \Sigma^t \in \mathbb{R}^{3\times 3}$
Initialize: $k = 1$
while *not converged* **do**
  Correspondence:
  $\boldsymbol{T}_{k-1}(\boldsymbol{b}_j) = \boldsymbol{t}_{k-1} + Real(\boldsymbol{q}_{k-1} \odot \boldsymbol{b}_j \odot \boldsymbol{q}^*_{k-1})$
  $\boldsymbol{c}_j = \text{FindClosestPoint}(\boldsymbol{T}_{k-1}(\boldsymbol{b}_j)),\ \boldsymbol{c}_j \in A,$
  State Update:
  $\boldsymbol{q}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{G}_k)\boldsymbol{q}_{k-1}$
  $\Sigma^{\boldsymbol{q}}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{G}_k)\Sigma^{\boldsymbol{q}}_{k-1}$
  $\boldsymbol{t}_k = \frac{1}{m}\left(\sum_{j=1}^m \boldsymbol{c}_j - Real\left(\boldsymbol{q}_k \odot \sum_{j=1}^m \boldsymbol{b}_j \odot \boldsymbol{q}^*_k\right)\right)$
  $\Sigma^{\boldsymbol{t}}_k = \boldsymbol{f}(\Sigma^{\boldsymbol{q}}_k, \Sigma^{\boldsymbol{b}_j}_k, \Sigma^{\boldsymbol{c}_j}_k)$
  $k = k + 1$

3.4. DQF with point and normal measurements

In this paper we extend our prior work to use surface normal measurements in addition to point measurements. The following equation relates the surface normals $\boldsymbol{n}^{b_j}$ in the sensor frame and the corresponding normal $\boldsymbol{n}^{a_j}$ in the model frame,

$$\begin{aligned}&\widetilde{\boldsymbol{n}}^{\boldsymbol{a}_j} = \boldsymbol{q} \odot \widetilde{\boldsymbol{n}}^{\boldsymbol{b}_j} \odot \boldsymbol{q}^* \quad j = 1, \ldots, m \\ &\Rightarrow \widetilde{\boldsymbol{n}}^{\boldsymbol{a}_j} \odot \boldsymbol{q} = \boldsymbol{q} \odot \widetilde{\boldsymbol{n}}^{\boldsymbol{b}_j} \\ &\Rightarrow \boldsymbol{J}_j \boldsymbol{q} = \boldsymbol{0}, \quad \text{where} \\ &\boldsymbol{J}_j = \begin{bmatrix} 0 & -(\boldsymbol{n}^{\boldsymbol{a}_j} - \boldsymbol{n}^{\boldsymbol{b}_j})^\text{T} \\ (\boldsymbol{n}^{\boldsymbol{a}_j} - \boldsymbol{n}^{\boldsymbol{b}_j}) & (\boldsymbol{n}^{\boldsymbol{a}_j} + \boldsymbol{n}^{\boldsymbol{b}_j})^\times \end{bmatrix}\end{aligned} \tag{7}$$

where $\boldsymbol{n}^{\boldsymbol{y}}$ is the surface normal at point $\boldsymbol{y} \in \mathbb{R}^3$. Note the similarity in the form of (7) and the derivation for point measurements (4). Instead of using a closest point rule to estimate the correspondence, we use a closest-oriented point rule. Such a rule has been widely used in literature with small variations such as the works of Glozman et al. (2001), Billings and Taylor (2014), and Pulli (1999).

The Kalman filter equations remain similar to those used for point measurements, but with a slight modification,

$$\boldsymbol{G} = [\boldsymbol{H}_1, \ldots, \boldsymbol{H}_\alpha, \boldsymbol{J}_1, \ldots, \boldsymbol{J}_m], \quad \boldsymbol{G} \in \mathbb{R}^{(\alpha+m)\times 4} \tag{8}$$

We shall refer to this variant of DQF that uses point and normal measurements as bDQFN. The algorithm for bDQFN is shown in Algorithm 4. If surface normal measurements are unavailable, then we use $\boldsymbol{G}$ as defined in (6) instead of (8). The rest of the steps remain the same. The complexity of bDQFN and bDQF is $O(m \log(n))$, similar to ICP.

4. Registration with sparse measurements

In this section, we describe our approach for registration with sparse measurements. Algorithm 5 and Figure 2 show the basic framework and the steps involved in our SPNR approach.

1. The algorithm is first initialized using an initial pose (see Figure 2(b)).
2. The current best pose is perturbed and p perturbed poses are obtained. In Figure 2, $p = 3$ is chosen. The amount of perturbation is reduced over the iterations. Refer to Sections 4.1 and 4.2 for more information on how to choose the number and amount of perturbation.

Algorithm 4: bDQF with point and normal measurements

Input:
$A = \{\boldsymbol{a}_i, \boldsymbol{n}^{\boldsymbol{a}_j} \in \mathbb{R}^3\},\ i = 1, 2, \ldots, n$
$B = \{\boldsymbol{b}_j, \boldsymbol{n}^{\boldsymbol{b}_j} \in \mathbb{R}^3\},\ j = 1, 2, \ldots, m$
Initial transformation: $\boldsymbol{q}_0 \in \mathbb{R}^4, \boldsymbol{t}_0 \in \mathbb{R}^3$,
Initial residual noise parameters: κ_0, σ_0^2
Output:
$\boldsymbol{q} \in \mathbb{R}^4, \boldsymbol{t} \in \mathbb{R}^3$ that aligns A and B
$\boldsymbol{S}^{\boldsymbol{q}} \in \mathbb{R}^{4\times 4}, \boldsymbol{S}^{\boldsymbol{t}} \in \mathbb{R}^{3\times 3}$
1. *Initialize*: $k = 1$, $\boldsymbol{q} \leftarrow \boldsymbol{q}_0$,$\boldsymbol{t} \leftarrow \boldsymbol{t}_0$, $\kappa \leftarrow \kappa_0$, $\sigma^2 \leftarrow \sigma_0^2$
2. **while** *not converged* **do**
3. **Correspondence**:
4. $\boldsymbol{T}_{k-1}(\boldsymbol{b}_j) = \boldsymbol{t}_{k-1} + \text{Real}(\boldsymbol{q}_{k-1} \odot \boldsymbol{b}_j \odot \boldsymbol{q}_{k-1}^*)$
5. $\boldsymbol{T}_{k-1}(\boldsymbol{n}_j^B) = \text{Real}(\boldsymbol{q}_{k-1} \odot \boldsymbol{n}_j^B \odot \boldsymbol{q}_{k-1}^*)$
6. $(\boldsymbol{c}_j, \boldsymbol{n}_j^c) = \text{FindClosestOrientedPoint}(\boldsymbol{T}_{k-1}(\boldsymbol{b}_j, \boldsymbol{n}_j^b), \sigma^2, \kappa)$,
7. **State Update**:
8. $\boldsymbol{q}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{G}_k)\boldsymbol{q}_{k-1}$
9. $\Sigma_k^{\boldsymbol{q}} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{G}_k)\Sigma_{k-1}^{\boldsymbol{q}}$
10. $\boldsymbol{t}_k = \frac{1}{m}\left(\sum_{j=1}^m \boldsymbol{c}_j - Real\left(\boldsymbol{q}_k \odot \sum_{j=1}^m \boldsymbol{b}_j \odot \boldsymbol{q}_k^*\right)\right)$
11. $\Sigma_k^{\boldsymbol{t}} = \boldsymbol{f}(\Sigma_k^{\boldsymbol{q}}, \Sigma_k^{\boldsymbol{b}_j}, \Sigma_k^{\boldsymbol{c}_j}, \Sigma_k^{\boldsymbol{n}^{\boldsymbol{b}_j}}, \Sigma_k^{\boldsymbol{n}^{\boldsymbol{c}_j}})$
12. **Update residual noise**:
13. $\sigma^2 = \frac{1}{3n}\sum_j \left\|\boldsymbol{c}_j - \boldsymbol{T}(\boldsymbol{b}_j)\right\|^2$
14. $\kappa = (\boldsymbol{b}_j, \boldsymbol{c}_j, \boldsymbol{n}^{\boldsymbol{b}_j}, \boldsymbol{n}^{\boldsymbol{c}_j})$
15. $k = k + 1$

Algorithm 5: Registration with sparse measurements

Input:
$A = \{\boldsymbol{a}_i, \boldsymbol{n}^{\boldsymbol{a}_i} \in \mathbb{R}^3\},\ i = 1, 2, \ldots, n$
$B = \{\boldsymbol{b}_j, \boldsymbol{n}^{\boldsymbol{b}_j} \in \mathbb{R}^3\},\ j = 1, 2, \ldots, m$
Initial transformation: $\boldsymbol{T}_0 \in SE(3)$
Output: $\boldsymbol{T} \in SE(3)$ that aligns A and B
1 *Initialize*: $\boldsymbol{T} \leftarrow \boldsymbol{T}_0$, $\Sigma \leftarrow \Sigma_0$ $k = 0$, $\varepsilon = \inf$
2 **while** $k < MaxIterations$ *OR* $\varepsilon > Threshold$ **do**
3 **Perturbation**:
4. $\widetilde{\boldsymbol{T}}_j$=Perturb$(\boldsymbol{T}_k, \mathcal{N}(0, \Sigma_k)), j = 1, \ldots, p$
5. **Evaluate Cost Function**:
6. $\boldsymbol{c}_i^j =$ FindClosestPoint$(\widetilde{\boldsymbol{T}}_j(\boldsymbol{b}_i))$, $\boldsymbol{c}_i^j \in A$
7. $O_j = \sum_{i=1}^m \parallel \widetilde{\boldsymbol{T}}_j(\boldsymbol{b}_i) - \boldsymbol{c}_i^j \parallel$
8. **Locally Optimal Estimate**:
9. $\widehat{\boldsymbol{T}}_k = \arg\min_{\widetilde{\boldsymbol{T}}_j} O_j$
10. $\boldsymbol{T}_{k+1}, \Sigma_{k+1}$=IMLOP$(A, B, \widehat{\boldsymbol{T}}_k)$ or bDQFN$(A, B, \widehat{\boldsymbol{T}}_k)$
11. $\boldsymbol{\epsilon}_k = \sum_{i=1}^m \parallel \boldsymbol{T}_k(\boldsymbol{b}_i) - \boldsymbol{c}_i \parallel$
12. **if** $\varepsilon_k < \varepsilon$ **then**
13. $\boldsymbol{T} = \boldsymbol{T}_k$, $\Sigma = \Sigma_k$, $\varepsilon = \varepsilon_k$
14. $k = k + 1$

3. A cost function is evaluated for each of the perturbed poses. The cost function is the sum of the closest distance between the point measurements and the geometric model: $O_j = \sum_{i=1}^m \parallel \widetilde{\boldsymbol{T}}_j(\boldsymbol{b}_i) - \boldsymbol{c}_i \parallel$, $j = 1, \ldots, p$, where $\widetilde{\boldsymbol{T}}_j \in SE(3)$ and $\boldsymbol{c}_i \in A$ is the closest point in the A. In this step, we use an approximate geometric model A' instead of A, to quickly evaluate the cost function. Depending on the format of the geometric model, several existing simplification techniques can be applied, such as those of Heckbert and Garland (1997), Garland and Heckbert (1997), Pauly et al. (2002), and Cignoni et al. (1998). For example, in this paper we work with a triangulated mesh model and use a quadric mesh simplification as shown in Garland and Heckbert (1997) (see Figures 2(c) and 7). The more approximate the geometry, the less the time taken to compute O_j but greater number of iterations for the overall algorithm. The complexity of evaluating the cost function is $O(m \log(n'))$, where $n' < n$ are the points in A'.
4. The pose $\widehat{\boldsymbol{T}} = \arg\min_{\widetilde{\boldsymbol{T}}_j} O_j$ is chosen as the initial guess for a locally optimal pose estimation using IMLOP or bDQFN. The complexity of obtaining locally optimal estimate is $O(m \log(n))$. In Section 5, we discuss the advantages and limitations of using deterministic methods such as ICP and IMLOP over probabilistic methods such as bDQF and bDQFN.
5. Steps 2–4 are repeated until convergence or up to a fixed number of iterations.

Thus, the overall complexity of SPNR is similar to ICP, $O(m \log(n))$. However, note that the actual run time of the algorithm would be greater than ICP/IMLP/bDQF/bDQFN, since the local optimizer is executed multiple number of times. In the worst-case scenario, when $A' = A$ and the algorithm is run for maximum number of iterations, the run time of SPNR would be $p \times MaxIterations$ times the runtime of the local optimizer (see Algorithm 5).

There are two variants of our SPNR approach: (1) deterministic SPNR (dSPNR); (2) probabilistic SPNR (pSPNR). These two variants are described in detail in the following.

4.1. Deterministic sparse point and normal registration (dSPNR)

In the dSPNR, IMLOP as described in Section 3.1 is used to find the locally optimal pose (Step 7 of SPNR in Algorithm 5). If only point measurements are available, then IMLOP simplifies to ICP and dSPNR trivially becomes equivalent to deterministic sparse point registration (dSPR) which was introduced in our prior work (Srivatsan et al., 2017a). There are three tunable parameters.

1. Number of perturbations: Perturbations help the optimizer move out of a local minimum. The higher the number of perturbations, the faster the convergence to the optimal estimate. But higher numbers of perturbations also imply higher computation times. In this work we choose $p = 10$ as the number of perturbations.
2. Amount of perturbation: The amount of perturbation helps balance exploration and exploitation. Higher perturbation encourages exploration while lower

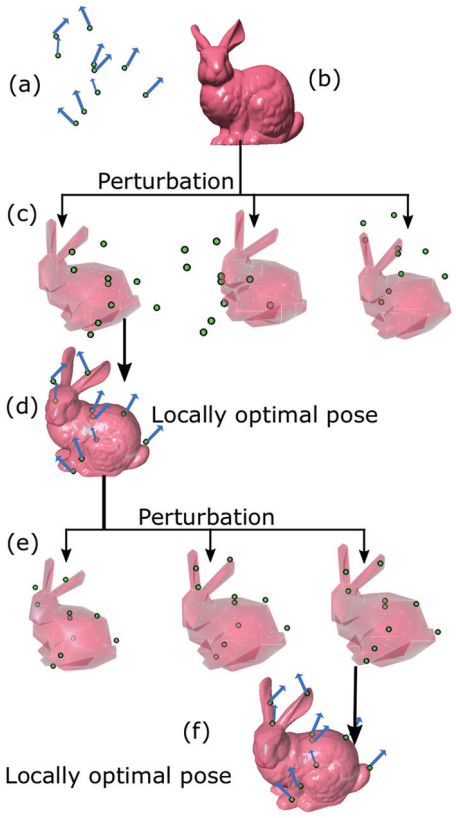

Fig. 2. The steps involved in two iterations of the registration approach with sparse measurements. The best pose estimate in each iteration is perturbed to obtain three pose particles. (a) Point and surface normal measurements. (b) Geometric model of the object. (c) Perturbed poses of a lower-resolution geometric model. The number of triangle vertices in the original model is 259,896, in comparison with 88 in the approximate model. (d) The best pose from the perturbed poses is selected and a locally optimal pose is obtained by using the original geometric model and ICP, IMLOP, bDQF, or bDQFN. (e) The current best pose is perturbed to obtain three new poses. The perturbation in this step is lower than the previous iteration. (f) The locally optimal pose is estimated. The steps are repeated until convergence.

perturbation encourages exploitation. We start with a high perturbation amount and decrease the perturbation over iterations. In this work, we set the initial perturbation in orientation to be drawn from a normal distribution with zero mean and a standard deviation of 30°. The initial perturbation in translation is drawn from a normal distribution with zero mean and a standard deviation of 10% of the maximum spatial dimension of object. The perturbation is decreased linearly until it is reduced to zero after a maximum of 30 iterations.

3. Termination criteria: The procedure can be terminated when the number of iterations reaches a set limit or when the root-mean-square (RMS) error between A and B is lower than a set threshold. In this work, we set the maximum number of iterations to 30 and the RMS error threshold to 0.5% of the maximum spatial dimension of object. In addition, the maximum number of iterations of each ICP step is set to 20.

Using a large number of perturbations and iterations, while decreasing the RMS error threshold improves the accuracy of the final results; at the cost of increased computation time. The choices of parameters made in this work for dSPNR are based on manual tuning over several standard datasets, and are not meant to exhibit any optimal behavior.

4.2. Probabilistic sparse point and normal registration (pSPNR)

In the pSPNR, bDQFN as described in Section 3.3 is used to find the locally optimal pose. As with dSPNR, when normal measurements are unavailable, pSPNR trivially becomes equivalent to probabilistic sparse point registration (pSPR), which was introduced in our prior work (Srivatsan et al., 2017a). There are two tunable parameters.

1. Number of perturbations: Since we use a Gaussian distribution-based filter, the number of perturbations is chosen to be equal to the number of sigma points. In the case of bDQFN, there are 15 sigma points. Unlike the dSPNR, the amount of perturbation need not be set manually, but can be chosen from a normal distribution with zero mean and standard deviation matching the standard deviation of the current state estimate.
2. Termination criteria: In this work, we set the maximum number of iterations to 12 and the RMS error threshold to 0.5% of the size of the object. The number of iterations in each bDQFN step is set to 50.

Using a large number of iterations in each bDQFN step while decreasing the RMS error threshold improves the accuracy of the final results, at the cost of increased computation time.

5. Simulation experiments

We performed a number of simulation experiments on standard shape datasets to systematically study the dSPNR and pSPNR. The results were compared with ICP, bDQF, UPF of Ma and Ellis (2004b), CPD of Myronenko and Song (2010), globally optimal ICP (GoICP) of Yang et al. (2013), and SparseICP of Bouaziz et al. (2013). For all the simulation experiments, the objects are scaled to fit in a cube of edge length 100 mm, for a fair comparison of the registration errors.[1]

5.1. Using point measurements versus point and normal measurements

In this section, we study the RMS error in registration using only point measurements versus using point and normal measurements. For this study we consider a Stanford bunny

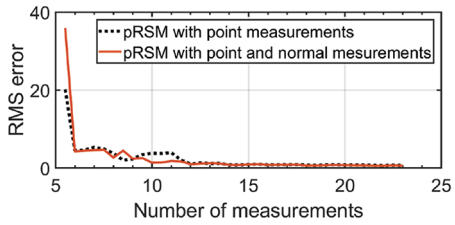

Fig. 3. Plot of RMS error versus number of measurements for pSPNR using point as well as point and normal measurements.

and sample points from the model of the bunny. Noise is added to each coordinate that is randomly sampled from a normal distribution $\mathcal{N}(0, 4 \times 10^{-2})$. We also add a noise to the normals at the chosen point locations. The normals are rotated about a random axis by an angle randomly chosen between $[0,\ 30]^\circ$. Following this, we apply a known transformation where the applied translation is uniformly drawn from $[-30,\ 30]$ mm and rotation about each primary axis is uniformly drawn from $[-30,\ 30]^\circ$. The experiment is repeated 100 times and the mean results are reported in Table 2. As shown in Figure 3, there is not much of a difference in terms of the RMS error with and without using normal measurements. However, using normal measurements produces a slight advantage in terms of accuracy when the number of measurements is less than 15. This trend is similar for dSPNR as well.

As shown in Table 2, when the number of measurements is less than 20, both dSPNR and pSPNR produce more accurate results when using normals as opposed to when using only point measurements. However, as we increase the number of point measurements, using the normals does not necessarily produce more accurate results. This is likely accentuated by the noise in the normal measurements.

It is also worth noting that the RMS error of dSPNR is generally lower than that of pSPNR, even though the computation time is lower for pSPNR. The difference in the RMS error between dSPNR and pSPNR decreases as we obtain more measurements. When the number of measurements is small (less than 20), dSPNR is faster than pSPNR. But when the number of measurements increases, pSPNR becomes faster than dSPNR.

5.2. Minimum number of measurements required for robust estimation

Different shapes need different numbers of points for reliable registration estimates. In theory, if the point correspondences are known, four points not lying on a plane are sufficient to unambiguously find the pose (Horn, 1987). If the correspondences are unknown, there may exist multiple valid solutions for the pose when only a small number of points are available (see Figure 4). While the authors are not aware of any prior work that describes the lower limit on the number of random points required to reliably estimate the pose, the works of Simon et al. (1995) (later extended by Ma and Ellis (2004a))comes closest to answering this question. Given the geometry of the object, Simon et al. (1995) found a small number of feature points in the frame of the geometric model, which when probed helps provide reliable registration estimates. But in order to probe these points, their locations need to be known in the robot frame. Thus, their approach produces good results only when the initial registration guess is close to the true registration.

In an attempt to empirically find the minimum number of point measurements required for reliable registration, we perform an experiment where p random points/normals from the model are selected. A known transformation is applied to these p points/normals. The applied transformation and RMS error are estimated using dSPNR. The applied transformation is parameterized by Cartesian coordinates $(x, y, z) \in \mathbb{R}^3$ and Euler angles $(\theta_x, \theta_y, \theta_z) \in \mathbb{R}^3$. Each translation parameter is uniformly drawn from $[-30,\ 30]$ mm and each orientation parameter is uniformly drawn from $[-30,\ 30]^\circ$. The experiment is repeated 100 times and the mean error is calculated. This process is repeated for different values of p, where $p \in \{4, 5, \ldots, 36\}$. We perform this experiment for several shapes namely, Bunny, Armadillo, Dragon, Happy Buddha, Lucy, Thai Statue (obtained from the Stanford Point Cloud library (Turk and Levoy, 2005)), Fertility (obtained from the AIM shape repository (AIM@SHAPE model repository, 2004)), femur bone[2], liver,[3] and pelvis bone.[4]

It is observed that some shapes such as the pelvis and Bunny require only 16 measurements, whereas others such

Table 2. Using point measurements versus point-normal measurements

		dSPNR		pSPNR	
Number of points	Number of normals	Time (s)	RMS (mm)	Time (s)	RMS (mm)
6	0	7.75	5.74	5.69	5.35
6	6	3.36	4.84	2.01	5.06
10	0	6.55	2.40	8.05	4.04
10	10	5.33	2.27	4.25	3.59
20	0	6.76	1.08	6.77	1.43
20	20	10.44	0.94	4.46	1.51
60	0	7.26	0.27	3.79	0.29
60	60	20.46	0.40	5.27	0.34

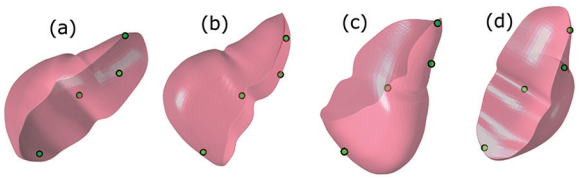

Fig. 4. Four different poses of the liver contain the same set of four green points.

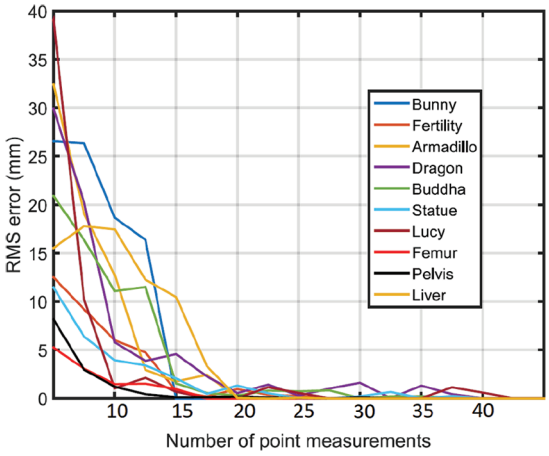

Fig. 5. Plot of RMS error versus number of points used for registration, when using dSPNR. For each integer element on the X axis, mean error is computed over 100 experiments. Most of the shapes considered need approximately 20 measurements for accurate registration.

as the Dragon require 36 measurements. Most of the shapes need approximately 20 measurements. Given a new shape, similar experiments can be run to empirically find the minimum number of measurements required for reliable registration estimates.

5.3. Robustness to noise

We repeated the experiment from the previous section for a the femur bone, with different amounts of noise added to the measurements. The dashed lines in Figure 6 show the mean error for dSPNR and pSPNR over 100 experiments versus the number of measurements used. The RMS error for both dSPNR and pSPNR decreases to zero after 12 and 18 measurements, respectively.

The experiments were repeated under identical conditions, with a noise uniformly drawn from $[-2, 2]$ mm added to each coordinate of the point measurements. Figure 6 shows that the mean error for dSPNR and pSPNR both converge to less than 2 mm after 20 measurements. The performance of both dSPNR and pSPNR are very similar, with and without measurement noise. Table 1 shows

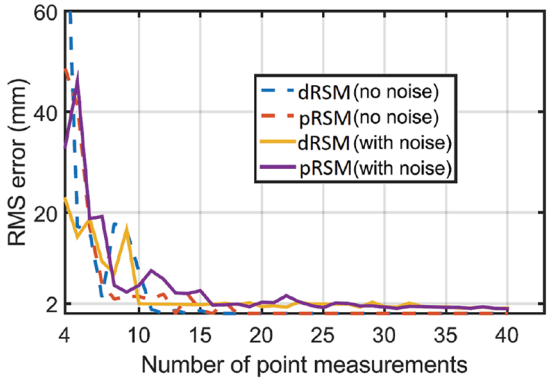

Fig. 6. Plot of the RMS error versus number of measurements used for dSPNR and pSPNR, with and without noise in the measurements. In the absence of noise, dSPNR takes 12 measurements and pSPNR takes 18 measurements to converge to zero RMS error. In the presence of a uniform noise of 2 mm, both pSPNR and dSPNR converge to an RMS error of 2 mm or less after 20 measurements.

the RMS error for dSPNR and pSPNR as well as the estimation time, for varying levels of noise, using 20 measurements and 100 measurements. When using 100 measurements, as expected, dSPNR, pSPNR, ICP, bDQF, CPD, and GoICP accurately estimate the registration. However, when using 20 measurements, dSPNR and pSPNR outperform all the other methods. UPF takes the most computation time compared with all the other methods because of the need to iterate over $\approx 2,000$ pose particles (Moghari and Abolmaesumi (2007) have also noted the high computation time of UPF). The output of GoICP is typically similar to ICP and sometimes better, but at the cost of high computation time. SparseICP performs the worst in all cases, because it is not developed for registering point clouds with such a small number of sparse points. Note that the results reported for GoICP and SparseICP are from a C++ implementation and, hence, their runtimes cannot be compared directly with those of the other methods that are implemented in MATLAB.

In Table 3 we present the results of a repeat of the experiment for a number of shapes from the Stanford point cloud dataset (Turk and Levoy, 2005), for the case of 5 mm noise and 20 measurements. We observe a trend similar to that for the case of femur with dSPNR and pSPNR outperforming other methods.

In this work, we do not present comparisons with other popular registration methods such as generalized ICP (GICP) of Segal et al. (2009), UKF-based registration of Moghari and Abolmaesumi (2007), DQF of Srivatsan et al. (2016b), iterative most likely point of Billings et al. (2015), etc. as those methods are not designed to work with fewer than 100 measurements and, hence, the comparison would be unfair (as observed with SparseICP).

Table 3. Registration in the presence of 5 mm noise on datasets from the Stanford point cloud library.

20 points	Bunny		Dragon		Armadillo		David		Buddha	
	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)
dSPNR	0.81	4.73	0.39	4.02	0.26	3.83	0.31	2.18	0.47	4.95
pSPNR	1.49	5.10	0.50	3.99	0.47	3.76	0.49	2.83	0.56	4.39
ICP	0.02	21.18	0.01	11.12	0.01	6.43	0.01	4.54	0.01	7.03
CPD	1.14	5.91	0.70	6.17	1.00	9.10	1.92	3.18	0.69	7.73
bDQF	0.09	14.23	0.04	7.16	0.03	5.36	0.04	4.42	0.04	8.14
UPF	141.28	36.02	108.24	24.39	112.76	25.45	123.78	31.26	114.44	37.19
GoICP	35.84	8.24	19.52	9.07	20.21	6.43	19.84	4.54	20.07	7.03
SparseICP	0.03	27.77	0.04	20.70	0.03	34.10	0.03	41.80	0.05	20.70

Table 4. Results for varying mesh resolution.

Number of faces	Time (s)		RMS error (mm)	
	dSPNR	pSPNR	dSPNR	pSPNR
14	0.17	0.10	0.17	0.17
148	0.28	0.37	0.17	0.18
1,488	0.43	046	0.17	0.17
14,896	0.61	1.14	0.17	0.18

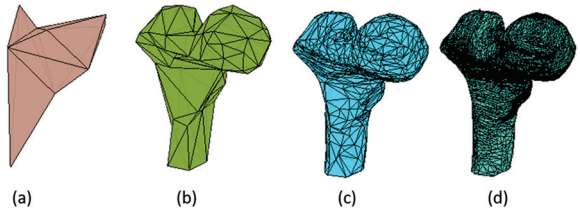

Fig. 7. Mesh model of the femur bone in various resolutions as generated using the "reducepatch" command of MATLAB®: (a) mesh with 14 faces; (b) mesh with 148 faces; (c) mesh with 1,488 faces; and (d) original mesh with 14,896 faces.

5.4. Effect of mesh model simplification

Step 3 of the SPNR approach involves evaluating an inexpensive cost function on an approximate geometric model. In order to study the effect of varying the resolution of approximation on the end result, in this section we consider a triangular mesh model of a femur bone with different levels of mesh approximation performed using the inbuilt MATLAB® command "reducepatch." The command "reducepatch" reduces the number of faces of the triangular mesh, while attempting to preserve the overall shape of the original object. We present the results of dSPNR and pSPNR in Table 4, for registration with 20 measurements. Table 4 shows the time taken and the RMS error for varying mesh resolutions (see Figure 7 for the mesh models are various resolutions). For reference, the original mesh model has 14,896 triangle faces (see Figure 7(d)). We observe that there is no significant difference in the RMS error, however, the time taken to converge is lowest for the mesh with the smallest number of faces. This study shows that approximating the original mesh model in Step 3 of SPNR to decrease the computation time does not degrade the registration results.

Table 5. Effect of changing the initial registration error.

Rotation error (degrees)	Translation error (mm)	Time (s)	RMS error (mm)
dSPNR			
10	10	1.51	0.35
20	20	1.74	0.56
30	30	1.80	3.04
pSPNR			
10	10	0.32	0.26
20	20	0.43	0.32
30	30	0.68	1.28

5.5. Effect of changing the initial registration error

Table 5 shows the RMS error and time taken for registration estimation using dSPNR and pSPNR for various initial registration errors. When the initial registration error is low, the RMS error is also low as expected. We observe that the computation time of dSPNR and pSPNR increases as the initial registration error increases. As observed in earlier experiments, the pSPNR is generally more accurate than dSPNR and takes less computation time. This study shows that if we were to use a pre-registration step as in the case of Glozman et al. (2001) or start from a good initial registration as in the case of Simon et al. (1995), the registration as estimated by our SPNR approach should be low as shown in Table 5.

5.6. Effect of changing the number of perturbations

When using the deterministic variant of SPNR, one parameter that needs tuning is the number of perturbations.

Table 6. Effect of varying number of perturbations in dSPNR.

Number of perturbations	Time (s)	RMS error (mm)
1	0.72	0.57
5	0.77	0.46
10	0.81	0.27
20	0.88	0.17

Perturbations can help escape local minima solutions and prevent premature convergence. However, perturbations can also delay convergence and increase computation time. To study the effect of these parameters, we take the earlier example of femur bone. We fix all the other parameters and vary the number of perturbations to study the RMS error and the computation time.

As shown in Table 6, increasing the number of perturbations results in an improvement in the registration accuracy. The computation time also increases as expected, with the increase in the number of perturbations. However, this increase in computation time is marginal compared with the improvement to the registration accuracy it brings.

6. Robot experiments

In order to test our approach to SPNR with real data, two robot experiments are presented: automatic probing and telemanipulated probing for registration.

6.1. Results for automatic probing

An experimental setup as shown in Figure 8 is used. The setup consists of a six-degree-of-freedom (6-DOF) robot Foxbot equipped with an ATI Nano17 force sensor at the end-effector with a probe attached to it. The object of interest is clamped in front of the robot and is probed using the strategies described in the following.

6.2. Point selection criteria

If an operator has visual information about the environment and telemanipulates the robot, then it is trivial to pick points on the object spread across the surface of the object. But if the robot is autonomously collecting point measurements, then it is critical to ensure that the points are randomly distributed over the surface. To this effect, two strategies have been developed in this work. For both strategies, first estimate the location and dimensions of a cuboid in the workspace of the robot, within which the object lies. The object can be probed from five faces of the cuboid (assuming the object rests on a table and cannot be probed from the bottom face). The guidelines for probing the object to obtain point measurements are as follows.

1. Choose a face of the cuboid at random, pick a point on this face at random, and probe along the direction joining the chosen point and the center of the bottom face of the cuboid (as shown in Figure 9(a)). Stop moving the robot once it makes contact with the object.
2. If the object is relatively flat (the smallest face of the cuboid is less than 30% of the largest face), then the previous strategy would result in most of the probed points lying on the face with the largest area. Thus, an alternate strategy is followed, where a random point is chosen on the face with largest area. The robot is moved in the direction of the surface normal of this face, until contact is made with the object (see Figure 9(b)).

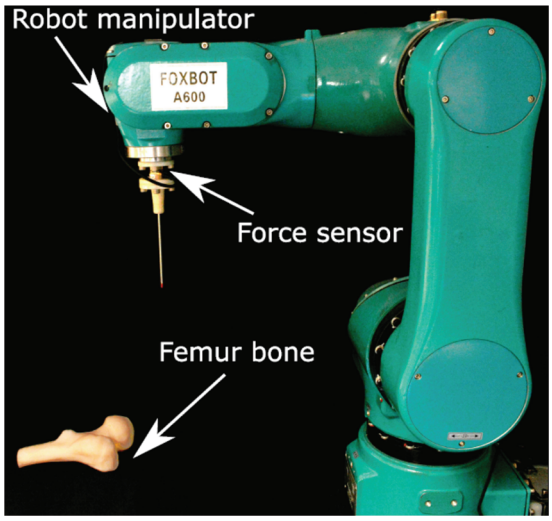

Fig. 8. Experimental setup consisting of a robot manipulator from Foxconn®, equipped with a force sensor. The object that is to be registered is clamped and held in place.

It is observed that such a strategy ensures that the points probed are spread evenly over the surface of the object. It is also observed that sometimes the robot might pass through holes in the object and make contact with the environment instead of the object. Such points are not considered in the computations in this work. Note that the probing strategies discussed here are only a general guideline, and the presented SPR approach would work fine if measurements are obtained using alternate strategies such as in Glozman et al. (2001); Javdani et al. (2013); Simon et al. (1995), or manually by a teleoperator as in Srivatsan et al. (2017b).

The objects chosen for this experiment are 3D printed models of femur bone, pelvis bone and Stanford bunny. The largest dimension for each of these objects is 100 mm. Using the information obtained from Figure 5, we collect 18, 20 and 20 points respectively for the femur, pelvis and bunny respectively.

The blue circles in Figure 10(b)–(d) show the initial guess for the location of the point measurements and the red circles show the location as estimated by dSPNR. Note that the estimated location of the points lie on the model of

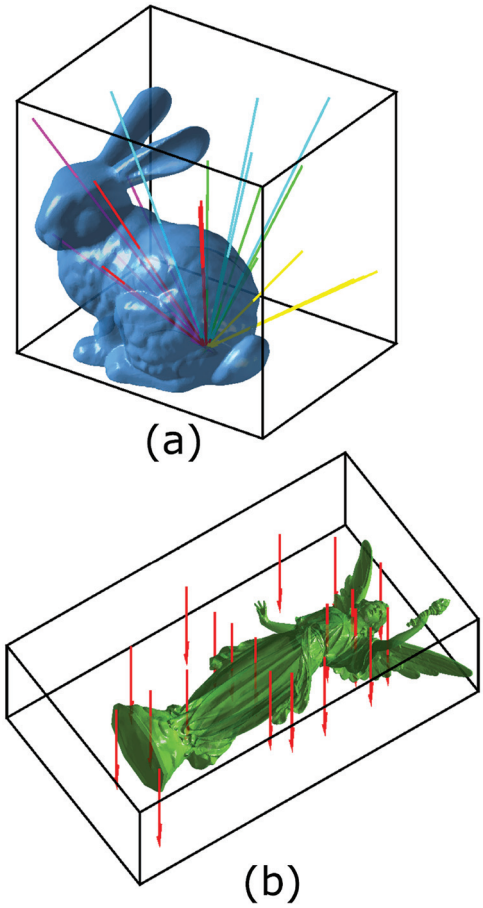

Fig. 9. A cuboid is selected in the workspace of the robot that conservatively estimates the location of the object. (a) Different probing paths for the robot are selected such that the probed points are spread across the surface of the object. The colors of the path show the face of the cuboid that the paths originate from. (b) Point collection strategy for relatively flat object. Some paths do not produce a point on the object. If the robot does not make contact with the object during the course of its path, then the point is not included in the registration.

the objects. The RMS errors for the locations estimated by dSPNR and pSPNR are listed in Table 7. We compare the results with ICP, CPD, and GoICP. As observed in the simulation experiments, dSPNR and pSPNR outperform the other methods. We do not report comparison results for other methods since they performed very poorly in our simulation experiments. For 20 point measurements, the RMS error for bunny is higher than the other two models as is consistent from our simulation experiments (see Figure 5).

We also performed experiments by probing a silicon phantom organ that was lubricated with glycerin (see Figure 11). The force vector detected directly provides the direction of the surface normal, since there is very little friction. This experiment provides both contact points and surface normal measurements to estimate the registration. We probed the organ at 20 random locations and obtained point and normal measurements at all these locations. Table 8 shows the results using different number of

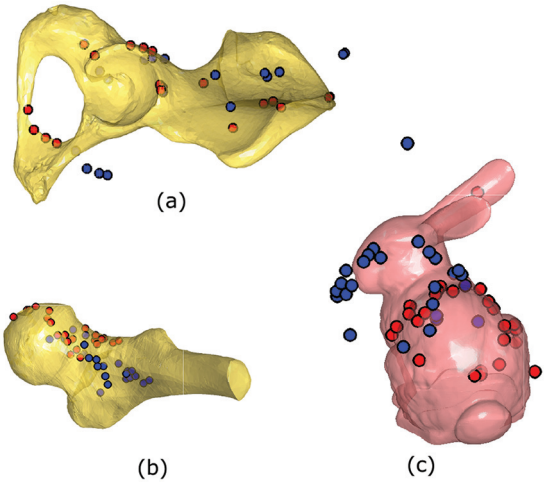

Fig. 10. Blue circles represent the initial location of the point measurements and red circles represent the registered location of the points. (a) Pelvis bone is probed at 18 points. (b) Femur bone is probed at 20 points. (c) Bunny is probed at 20 points. dSPNR and pSPNR accurately register the points to the model of the objects.

Table 7. Experimental results for automatic probing.

	Femur		Pelvis		Bunny	
	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)
dSPNR	1.5	2.17	1.56	1.38	1.76	5.00
pSPNR	3.38	2.21	3.01	2.33	3.95	4.91
ICP	0.05	4.41	0.04	17.12	0.02	9.37
CPD	0.06	16.27	0.04	6.30	0.06	28.09
GoICP	37.49	4.41	40.13	15.54	30.29	9.34

measurements. The RMS error for SPNR with 20 point measurements is higher than SPNR with 15 point and surface normal measurements. This clearly shows that using surface normal measurements helps achieve more accurate registration with fewer probings. Further, we also observe that the RMS error of dSPNR is higher than pSPNR. This is due to the fact that by design the probabilistic approach is better at handling noise than the deterministic approach.

6.3. Results for telemanipulated probing

The experimental setup consists of a 3D print of the femur model used for our simulated experiments (see Figure 12). The femur model is modified to have four fiducials as shown in Figure 13(a). These fiducials are used to generate the ground truth for the registration. The femur is clamped onto the table along with a robot arm: the patient side manipulator of the daVinci Research Kit (dVRK), which is a research version of the daVinci surgical system, routinely used for robot-assisted surgery worldwide (Kazanzides et al., 2014). The robot is telemanipulated by a user using a

Table 8. Using point measurements vs point-normal measurements.

		dSPNR		pSPNR	
Number of normals	Number of points	Time (s)	RMS (mm)	Time (s)	RMS (mm)
15	0	0.27	3.86	1.43	1.59
15	15	4.30	3.33	2.00	1.23
20	0	0.39	4.26	1.69	1.51
20	20	6.38	2.96	1.94	1.13

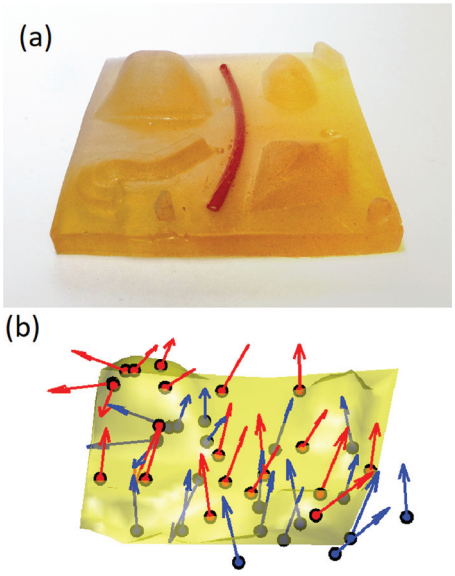

Fig. 11. (a) A phantom organ with lubricated surface for automatic probing-based registration using point and normal measurements. (b) The phantom organ is probed at 20 locations. Blue circles represent the initial location of the point measurements, blue arrows represent the corresponding surface normals, and red circles and red arrows represent the location of the points and surface normals after registration using pSPNR.

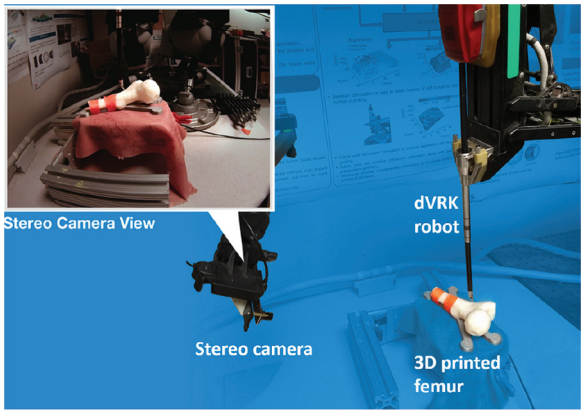

Fig. 12. Experimental setup for the telemanipulation experiments consists of a phantom organ clamped to the table, a stereo camera to provide the user direct image feedback, and a daVinci robot arm to probe the organ. On the top left is the view from the camera that the used by the teleoperator to probe the organ.

direct visual feedback from a stereo camera, to probe points along the surface of the femur, as well as the four fiducial points for ground truth registration (see Figure 12). In all, 25 points were collected, in addition to the four ground truth registration points. These points were chosen to roughly cover the area of the femur model that the operator could reach with the telemanipulation system. We estimate the contact using current-sensing on the patient side manipulator. Although this measurement is noisy and somewhat inaccurate, it is sufficient for detecting static point contacts. It also represents a realistic surgical scenario where the robot would likely not be equipped with an external force sensor, unlike the robot experiment presented earlier. At each point, robot positions were collected using the robot's forward kinematics. Since correspondence for the four ground truth points was known, the approach of Horn (1987) was used to find the ground truth registration from the fiducial points. The remaining 25 points were used to estimate the registration using dRSM as well as pRSM (see Figure 13(b) for results). Results are listed in Table 9.

Table 9. Experimental results for telemanipulated probing.

	10		15		20		25	
	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)	Time (s)	RMS (mm)
dSPNR	2.26	3.95	2.14	3.82	2.30	3.09	2.27	2.29
pSPNR	3.02	36.40	4.34	3.93	3.92	3.81	4.54	3.62
ICP	0.05	3.83	0.05	5.09	0.05	5.93	0.05	4.41
CPD	0.99	13.58	1.21	14.14	1.43	16.13	1.61	16.27
GoICP	19.96	3.83	19.62	4.82	19.84	5.35	19.62	4.41

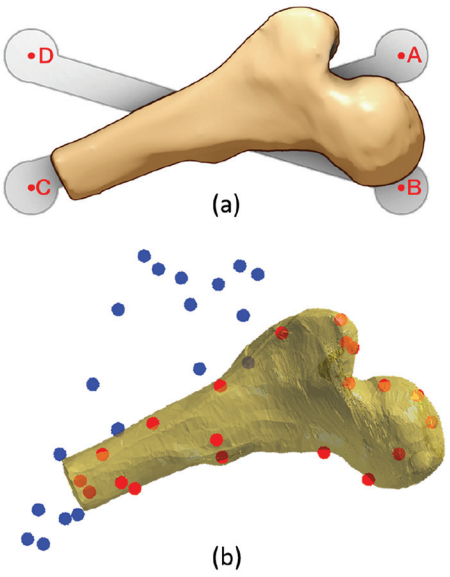

Fig. 13. (a) There are four fiducials A, B, C, and D that are rigidly attached to the femur at known locations. These are used to find the ground truth registration. (b) Femur bone is probed at 20 points. dSPNR and pSPNR accurately register the points to the model of the objects. Blue circles represent the initial location of the point measurements and red circles represent the registered location of the points.

7. Discussions and future work

In this work, an approach for robust registration with a small number of sparse measurements (SPNR) was developed. The approach can be implemented in a deterministic manner (dSPNR) or a probabilistic manner (pSPNR). The dSPNR is computationally faster than pSPNR. But the pSPNR has fewer parameters to tune than dSPNR, and hence is easier to use. Furthermore, the pSPNR has the added advantage over dSPNR of providing the uncertainty in the registration estimate. We also provide guidelines for choosing the tuning parameters for each algorithm and systematically study the effect of choosing the right parameter values on the registration estimate. Another contribution of this work is the development of a batch processing variant of the dual quaternion filter, which can use both normal and point measurements to provide registration estimates. We observe that using normal measurements can improve the registration accuracy when there are a small number of measurements.

Through simulations and robot experiments, both dSPNR and pSPNR are found to be robust and accurate compared with state-of-the-art methods. Even in the presence of noise, our approach accurately estimates the registration compared with popular deterministic and probabilistic approaches for registration. The computation time was around 1.5 s for registering using 20 measurements for most shapes. A C++ implementation would greatly reduce the computation time and allow for easy integration to other programs such as ROS and MATLAB, and this is a part of our future work. Through a number of simulations, it is empirically found that most shapes require approximately 20 measurements for reliable registration. Future work would explore a more theoretical approach for finding the lower bound on the number of random points required for registration. Future work would also involve using color intensity, stiffness, and other available heuristics to guide the measurement collection and registration.

Funding

This work was supported by the NRI (large grant IIS-1426655) and the Center for Machine Learning and Health, Carnegie Mellon University.

ORCID iD

Rangaprasad Arun Srivatsan https://orcid.org/0000-0002-0863-5524

Notes

1. All the computations are carried using MATLAB R2015a software from MathWorks, running on a ThinkPad T450s computer with 8 GB RAM and Intel i7 processor.
2. See https://grabcad.com/library/femur-bone
3. See https://grabcad.com/library/liver
4. See http://www.thingiverse.com/thing:1645117

References

Agamennoni G, Fontana S, Siegwart RY and Sorrenti DG (2016) Point clouds registration with probabilistic data association. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4092–4098.

AIM@SHAPE model repository (2004) Fertility point cloud scan. http://visionair.ge.imati.cnr.it/ontologies/shapes/releases.jsp.

Arun KS, Huang TS and Blostein SD (1987) Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5: 698–700.

Audette MA, Ferrie FP and Peters TM (2000) An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis* 4(3): 201–217.

Banerjee A, Dhillon IS, Ghosh J and Sra S (2005) Clustering on the unit hypersphere using von Mises–Fisher distributions. *Journal of Machine Learning Research* 6: 1345–1382.

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 (2015) *Probabilistic Feature-Based Registration for Interventional Medicine*. PhD Thesis, Johns Hopkins University.

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.

Bouaziz S, Tagliasacchi A and Pauly M (2013) Sparse iterative closest point. In: *Proceedings of the Eleventh Eurographics/*

ACMSIGGRAPH Symposium on Geometry Processing. Eurographics Association, pp. 113–123.

Chhatpar SR and Branicky MS (2005) Particle filtering for localization in robotic assemblies with position uncertainty. In: *IROS*. pp. 3610–3617.

Cignoni P, Montani C and Scopigno R (1998) A comparison of mesh simplification algorithms. *Computers and Graphics* 22(1): 37–54.

Eckart B, Kim K, Troccoli A, Kelly A and Kautz J (2015) MLMD: Maximum likelihood mixture decoupling for fast and accurate point cloud registration. In: *2015 International Conference on 3D Vision (3DV)*. IEEE, pp. 241–249.

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*. Berlin: Springer, pp. 234–241.

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.

Gadeyne K and Bruyninckx H (2001) Markov techniques for object localization with force-controlled robots. In: *International Conference on Advanced Robotics*.

Garland M and Heckbert PS (1997) Surface simplification using quadric error metrics. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press/Addison-Wesley, pp. 209–216.

Gelfand N, Mitra NJ, Guibas LJ and Pottmann H (2005) Robust global registration. In: *Symposium on Geometry Processing*, Vol. 2, p. 5.

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

Glozman D, Shoham M and Fischer A (2001) A surface-matching technique for robot-assisted registration. *Computer Aided Surgery* 6(5): 259–269.

Gold S, Rangarajan A, Lu CP, Pappu S and Mjolsness E (1998) New algorithms for 2D and 3D point matching: Pose estimation and correspondence. *Pattern Recognition* 31(8): 1019–1031.

Granger S and Pennec X (2002) Multi-scale EM-ICP: A fast and robust approach for surface registration. In: *European Conference on Computer Vision*. Berlin: Springer, pp. 418–432.

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

Hebert P, Howard T, Hudson N, Ma J and Burdick JW (2013) The next best touch for model-based localization. In: *Proceedings of ICRA*, pp. 99–106.

Heckbert PS and Garland M (1997) Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon university, Pittsburgh, PA, School of Computer Science.

Horn BK (1987) Closed-form solution of absolute orientation using unit quaternions. *The Journal of the Optical Society of America A* 4(4): 629–642.

Horowitz MB, Matni N and Burdick JW (2014) Convex relaxations of SE(2) and SE(3) for visual pose estimation. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1148–1154.

Hsiao K and Kaelbling LP (2010) Task-driven tactile exploration. In: *Robotics: Science and Systems*.

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*.

Javdani S, Klingensmith M, Bagnell JA, Pollard NS and Srinivasa SS (2013) Efficient touch based localization through submodularity. In: *International Conference on Robotics and Automation*. IEEE, pp. 1828–1835.

Jian B and Vemuri BC (2005) A robust algorithm for point set registration using mixture of Gaussians. In: *Tenth IEEE International Conference on Computer Vision, 2005 (ICCV 2005)*, Vol. 2. IEEE, pp. 1246–1251.

Jiang J, Cheng J and Chen X (2009) Registration for 3-D point cloud using angular-invariant feature. *Neurocomputing* 72(16–18): 3839–3844.

Kazanzides P, Chen Z, Deguet A, Fischer GS, Taylor RH and DiMaio SP (2014) An open-source research kit for the da Vinci Surgical System. In: *ICRA*. IEEE, pp. 6434–6439.

Li Y, Wang G, Ji X, Xiang Y and Fox D (2018) DeepIM: Deep iterative matching for 6D pose estimation. arXiv preprint arXiv:1804.00175.

Luck J, Little C and Hoff W (2000) Registration of range data using a hybrid simulated annealing and iterative closest point algorithm. In: *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE, pp. 3739–3744.

Luong HQ, Vlaminck M, Goeman W and Philips W (2016) Consistent ICP for the registration of sparse and inhomogeneous point clouds. In: *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*. IEEE, pp. 262–267.

Ma B and Ellis RE (2003) Robust registration for computer-integrated orthopedic surgery: Laboratory validation and clinical experience. *Medical Image Analysis* 7: 237–250.

Ma B and Ellis RE (2004a) Spatial-stiffness analysis of surface-based registration. In: *MICCAI*, pp. 623–630.

Ma B and Ellis RE (2004b) Surface-based registration with a particle filter. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Berlin: Springer, pp. 566–573.

Makadia A, Patterson A and Daniilidis K (2006) Fully automatic registration of 3D point clouds. In: *CVPR*, Vol. 1. IEEE, pp. 1297–1304.

Maron H, Dym N, Kezurer I, Kovalsky S and Lipman Y (2016) Point registration via efficient convex relaxation. *ACM Transactions on Graphics* 35(4): 73.

Mitash C, Boularias A and Bekris K (2018) Robust 6D object pose estimation with stochastic congruent sets. *arXiv preprint arXiv:1805.06324*.

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, p. 76231A.

Myronenko A and Song X (2010) Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12): 2262–2275.

Pauly M, Gross M and Kobbelt LP (2002) Efficient simplification of point-sampled surfaces. In: *Conference on Visualization'02*. pp. 163–170.

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.

Penney GP, Edwards PJ, King AP, Blackall JM, Batchelor PG and Hawkes DJ (2001) A stochastic iterative closest point algorithm (stochastICP). In: *MICCAI*, pp. 762–769.

Petrovskaya A and Khatib O (2011) Global localization of objects via touch. *IEEE Transactions on Robotics* 27(3): 569–585.

Phillips JM, Liu R and Tomasi C (2007) Outlier robust ICP for minimizing fractional RMSD. In: *Sixth International Conference on 3-D Digital Imaging and Modeling (3DIM'07)*. IEEE, pp. 427–434.

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

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.

Rusu RB, Blodow N and Beetz M (2009) Fast point feature histograms (FPFH) for 3D registration. In: *IEEE International Conference on Robotics and Automation, 2009 (ICRA'09)*, pp. 3212–3217.

Saund B, Chen S and Simmons R (2017) Touch based localization of parts for high precision manufacturing. In: *International Conference on Robotics and Automation*. IEEE.

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

Seixas FL, Ochi LS, Conci A and Saade DM (2008) Image registration using genetic algorithms. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York: ACM Press, pp. 1145–1146.

Simon DA, Hebert M and Kanade T (1995) Techniques for fast and accurate intrasurgical registration. *Journal of Image Guided Surgery* 1(1): 17–29.

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: *International Conference on Robotics and Automation*, pp. 924–930.

Srivatsan RA and Choset H (2016) Multiple start branch and prune filtering algorithm for nonconvex optimization. In: *WAFR*.

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

Srivatsan RA, Vagdargi P and Choset H (2017a) Sparse point registration. In: *18th International Symposium on Robotics Research*.

Srivatsan RA, Vagdargi P, Zevallos N and Choset H (2017b) Multimodal registration using stereo imaging and contact sensing. In: *Robotics: Science and Systems, Workshop on 'Revisiting Contact - Turning a problem into a solution'*.

Srivatsan RA, Xu M, Zevallos N and Choset H (2018) Probabilistic pose estimation using a Bingham distribution-based linear filter. *The International Journal of Robotics Research* 37(13-14): 1610–1631.

Tam GK, Cheng ZQ, Lai YK, et al. (2013) Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics* 19(7): 1199–1217.

Tsin Y and Kanade T (2004) A correlation-based approach to robust point set registration. In: *European Conference on Computer Vision*. Berlin: Springer, pp. 558–569.

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

Wachowiak MP, Smolková R, Zheng Y, Zurada JM and Elmaghraby AS (2004) An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation* 8(3): 289–301.

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

Xiang Y, Schmidt T, Narayanan V and Fox D (2017) PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199.

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

Yuan W, Khot T, Held D, Mertz C and Hebert M (2018) PCN: Point Completion Network. arXiv preprint arXiv:1808.00671.