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
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 (≈30° 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), and Walker 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; Tan et al., 2013), incorporate uncertainty in measurements (Esteépar 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

Fig. 1. (a) Geometric model of the object. (b) Point measurements in the sensor frame. (c) Point measurements registered to the geometric model.
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 Vernieri, 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
parameters in each iteration which results in each measurement being perturbed in a structured manner.

It is also worth mentioning that there are 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, using a network trained on similar point cloud datasets. 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, \( A = \{a_i\}, \quad a_i \in \mathbb{R}^3, \quad i = 1, \ldots, n \), are \( n \) points on the geometric model of the object and \( B = \{b_j\}, \quad b_j \in \mathbb{R}^3, \quad j = 1, \ldots, m \), are \( m \) points obtained using sensor measurements. Let \( T \in SE(3) \) be the transformation that aligns \( A \) and \( B \). The point \( c_j \) is the current best estimate for the model point that corresponds to sensor point \( b_j \), and is obtained using a closest point rule:

\[
c_j = \arg \min_{c \in A} \| c - T(b_j) \|^2
\]

where \( T(b_j) = R b_j + t, \quad R \in \mathbb{R}^{3 \times 3} \) is the rotation matrix and \( t \in \mathbb{R}^3 \) 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 \( O(nn) \). However, using a kd-tree search to estimate correspondence reduces the complexity to \( O(m \log(n)) \).

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

<table>
<thead>
<tr>
<th></th>
<th>No noise</th>
<th>2 mm noise</th>
<th>5 mm noise</th>
<th>10 mm noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>RMS (mm)</td>
<td>Time (s)</td>
<td>RMS (mm)</td>
</tr>
<tr>
<td>dSPNR</td>
<td>0.08</td>
<td>0.44</td>
<td>0.46</td>
<td>0.56</td>
</tr>
<tr>
<td>pSPNR</td>
<td>0.07</td>
<td>1.00</td>
<td>0.99</td>
<td>1.26</td>
</tr>
<tr>
<td>ICP</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>CPD</td>
<td>0.87</td>
<td>0.81</td>
<td>0.75</td>
<td>0.94</td>
</tr>
<tr>
<td>bDQF</td>
<td>0.05</td>
<td>0.09</td>
<td>0.09</td>
<td>0.1</td>
</tr>
<tr>
<td>UPF</td>
<td>161.77</td>
<td>159.83</td>
<td>242.38</td>
<td>248.56</td>
</tr>
<tr>
<td>GoICP</td>
<td>20.29</td>
<td>22.25</td>
<td>21.71</td>
<td>21.81</td>
</tr>
<tr>
<td>SparseICP</td>
<td>0.05</td>
<td>0.06</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Time (s)</td>
<td>RMS (mm)</td>
<td>Time (s)</td>
<td>RMS (mm)</td>
</tr>
<tr>
<td>dSPNR</td>
<td>0.03</td>
<td>0.51</td>
<td>0.53</td>
<td>0.82</td>
</tr>
<tr>
<td>pSPNR</td>
<td>0.13</td>
<td>2.53</td>
<td>3.10</td>
<td>2.88</td>
</tr>
<tr>
<td>ICP</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>CPD</td>
<td>1.29</td>
<td>1.35</td>
<td>1.39</td>
<td>1.62</td>
</tr>
<tr>
<td>bDQF</td>
<td>0.08</td>
<td>0.19</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>UPF</td>
<td>1891.1</td>
<td>1843.3</td>
<td>1594.0</td>
<td>1621.5</td>
</tr>
<tr>
<td>GoICP</td>
<td>20.28</td>
<td>20.06</td>
<td>20.18</td>
<td>39.75</td>
</tr>
<tr>
<td>SparseICP</td>
<td>0.15</td>
<td>0.11</td>
<td>0.11</td>
<td>0.11</td>
</tr>
</tbody>
</table>

### 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.
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,

\[
(c_j, n^o_j) = \arg \min_{(c, n^o) \in A} \frac{1}{2} \sigma^2 - \kappa (n^o)^T R b
\]

where \(\sigma^2\) and \(\kappa\) are residual match errors. The expression for \(\sigma^2\) is simply estimated as the mean square distance between the matches divided by the spatial dimensionality, \(\sigma^2 = \frac{1}{4n} \sum_i \|c_i - T(b_i)\|^2\). The residual error in surface normals is estimated using an approximation as derived by Banerjee et al. (2005), \(\kappa = h(b_i, c_j, n^b_i, n^o_j)\), where the expression of \(\kappa\) 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.

### 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.

**Algorithm 1. ICP Update**

**Input:**
- \(A = \{a_i \in \mathbb{R}^3\}, i = 1, 2, \ldots, n\)
- \(B = \{b_j \in \mathbb{R}^3\}, j = 1, 2, \ldots, m\)
- Initial transformation: \(T_0 \in SE(3)\)

**Output:** \(T \in SE(3)\) that aligns \(A\) and \(B\)

1. Initialize: \(T \leftarrow T_0\)
2. while not converged do
   1. **Correspondence:** \(c_j = \text{FindClosestPoint}(T(b_j))\), \(c_j \in A\)
   2. **Minimization:** \(T = \arg \min_T \frac{1}{m} \sum_{j=1}^m \|c_j - T(b_j)\|^2\)

**Algorithm 2. IMLOP Update**

**Input:**
- \(A = \{a_i \in \mathbb{R}^3\}, i = 1, 2, \ldots, n\)
- \(B = \{b_j \in \mathbb{R}^3\}, j = 1, 2, \ldots, m\)
- Initial transformation: \(T_0 \in SE(3)\)
- Initial residual noise parameters: \(\kappa_0, \sigma_0^2\)

**Output:** \(T \in SE(3)\) that aligns \(A\) and \(B\)

1. Initialize: \(T \leftarrow T_0\), \(\kappa \leftarrow \kappa_0\), \(\sigma^2 \leftarrow \sigma_0^2\)
2. while not converged do
   1. **Correspondence:** \((c_j, n^o_j) \in A\) = FindClosestOrientedPoint\((T(b_j), R(n^b_j), \sigma^2, \kappa)\)
   2. **Minimization:**
      \(T = \arg \min_T \frac{1}{m} \sum_{j=1}^m \|c_j - T(b_j)\|^2 - \kappa \sum_{j=1}^m (n^o_j)^T R b_j\)
   3. **Update residual noise:** \(\sigma^2 = \frac{1}{m} \sum_{j=1}^m \|c_j - T(b_j)\|^2\)
   4. \(\kappa = (b_j, c_j, n^b_j, n^o_j)\)

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 \(T\) is parameterized using the unit dual quaternion \(x = (q, d)^T\), where \(q \in \mathbb{R}^4\) is a unit quaternion that parameterizes the rotation, \(d = (0, t)^T \circ q\)/2, where \(\circ\) is the quaternion multiplication operator.

In our prior work (Srivatsan et al., 2016b), we used a pair of point measurements \(b_j\) and \(b_{j+1}\) in each iteration to update the registration estimate. Let \(c_j\) and \(c_{j+1}\) be the best estimate for the points on the model that correspond to \(b_j\) and \(b_{j+1}\), 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

\[
\tilde{c}_j = q \odot \tilde{b}_j \odot q^* + \tilde{t} \quad (1)
\]

\[
\tilde{c}_{j+1} = q \odot \tilde{b}_{j+1} \odot q^* + \tilde{t} \quad (2)
\]

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

\[
H q = 0, \quad H \in \mathbb{R}^{4 \times 4} \quad (3)
\]

\[
H = \begin{bmatrix}
0 & -(c_{j+1} - b_{j+1})^T \\
(c_{j+1} - b_{j+1}) & (c_{j+1} - b_{j+1}) \times
\end{bmatrix}
\]

\[
t = \frac{c_{j+1}}{2} - \text{Real}\left(q \odot \frac{b_{j+1}}{2} \odot q^*ight) \quad (5)
\]

where \(c_{j+1} = c_j - c_{j+1}, \quad b_{j+1} = b_j - b_{j+1}, \quad j = 1, 3, \ldots, 2\alpha - 1, \quad \alpha = \left\lfloor \frac{m}{2} \right\rfloor, \quad m = \text{the total number of measurements},\)
$q^*$ is the conjugate quaternion and $[ ]^T$ is the operator that converts a vector to a skew-symmetric matrix and $\text{Real}(v) = [v_2, v_3, v_4]^T$ for all $v = [v_1, v_2, v_3, v_4]^T \in \mathbb{R}^4$. 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 $q$ from (4) does not directly give us the rotation. We instead use a Kalman filter to estimate $q$ iteratively. The update equations for the Kalman filter are

$$q_k = q_{k-1} - K_k H_k q_{k-1}$$

$$\Sigma_k^q = (I - K_k H_k) \Sigma_{k-1}^q (I - K_k H_k)^T + Q_k$$

where

$$K_k = \Sigma_k^q H_k^T (H_k \Sigma_{k-1}^q H_k^T + Q_k)^{-1}$$

where $\Sigma_k^q$ is the uncertainty in the quaternion $q_k$ and $Q_k$ is the uncertainty in $z_k$, where $z_k = H_k q_{k-1}$. The translation vector $t_k$ is obtained from $q_k$ using (5). For the sake of brevity, the derivation for $Q_k = g(\Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q)$ and $\Sigma_k^q = f(\Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q, \Sigma_k^q)$ representing the uncertainty in the translation, are omitted here. The expressions along with their derivation can be obtained from Section IV of Srivatsan et al. (2016b).

The standard implementation of DQF requires $\approx 100$ 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

$$Gq = 0, \quad G \in \mathbb{R}^{m \times 4}, \quad \text{where } G = [H_1, \ldots, H_n]$$

(6)

Here $H_i$ is as defined in (4). The update equations of the filter remain the same. The algorithm for bDQF is shown in Algorithm 3.

### 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 $\vec{n}^b$ in the sensor frame and the corresponding normal $\vec{n}^w$ in the model frame,

$$\vec{n}^w = q \odot \vec{n}^b \odot q^*, \quad j = 1, \ldots, m$$

$$\Rightarrow \vec{n}^w \odot \vec{q} = q \odot \vec{n}^b$$

$$\Rightarrow J_q = 0, \quad \text{where}$$

$$J_q = \begin{bmatrix} 0 & -(\vec{n}_i - \vec{n}_j^b)^T \\ (\vec{n}_i - \vec{n}_j^b)^T & (\vec{n}_i + \vec{n}_j^b)^T \end{bmatrix}$$

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,

$$G = [H_1, \ldots, H_n, J_1, \ldots, J_m], \quad G \in \mathbb{R}^{(n+m) \times 4}$$

(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 $G$ 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:**

\[ A = \{a_i, n_i \in \mathbb{R}^3 \}, \quad i = 1, 2, \ldots, n \]

\[ B = \{b_j, n_j \in \mathbb{R}^3 \}, \quad j = 1, 2, \ldots, m \]

Initial transformation: \( q_0 \in \mathbb{R}^4, t_0 \in \mathbb{R}^3 \), Initial residual noise parameters: \( \kappa_0, \sigma_0^2 \)

**Output:**

\[ q \in \mathbb{R}^4, t \in \mathbb{R}^3 \] that aligns \( A \) and \( B \)

![Algorithm 4](image)

### Algorithm 5: Registration with sparse measurements

**Input:**

\[ A = \{a_i, n_i \in \mathbb{R}^3 \}, \quad i = 1, 2, \ldots, n \]

\[ B = \{b_j, n_j \in \mathbb{R}^3 \}, \quad j = 1, 2, \ldots, m \]

Initial transformation: \( T_0 \in SE(3) \)

**Output:** \( T \in SE(3) \) that aligns \( A \) and \( B \)

1. Initialize: \( k = 1, q \leftarrow q_0, t \leftarrow t_0, \kappa \leftarrow \kappa_0, \sigma^2 \leftarrow \sigma_0^2 \)
2. while not converged do
3. Correspondence:
4. \[ T_{k-1}(b_j) = t_{k-1} + \text{Real}(q_{k-1} \odot b_j \odot q_{k-1}^*) \]
5. \[ T_k(n_j) = \text{Real}(q_k \odot n_j^* \odot q_k^*) \]
6. \( (c_j, n_j^*) = \text{FindClosestPoint}(T_{k-1}(b_j), n_j) \)
7. State Update:
8. \[ q_k = \left( I - K_0 G_0 \right) q_{k-1} \]
9. \[ S_k = \left( I - K_0 G_0 \right) S_{k-1} \]
10. \[ t_k = \frac{1}{n} \sum_{j=1}^{n} \left[ c_j - \text{Real}(q_k \odot c_j) \right] \]
11. \[ S_k = \frac{1}{n} \sum_{j=1}^{n} \left( c_j^* \odot \left( S_{j-1} \odot c_j \odot S_{j-1}^* \right) \right) \]
12. Update residual noise:
13. \[ \sigma^2 = \frac{1}{m} \sum_{j=1}^{m} || c_j - T(b_j) ||^2 \]
14. \( k = k + 1 \)

### 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 dSPN 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...
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 $30^\circ$. 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.

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
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 \((\theta_x, \theta_y, \theta_z) \in \mathbb{R}^3\).

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

Table 2. Using point measurements versus point-normal measurements.
as the Dragon require 36 measurements. Most of the shapes need approximately 20 measurements for accurate registration.

5.3. Robustness to noise

We repeated the experiment from the previous section for a 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 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 $2,000$ 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).
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.

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.
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)).

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.

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

<table>
<thead>
<tr>
<th>Number of perturbations</th>
<th>Time (s)</th>
<th>RMS error (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.72</td>
<td>0.57</td>
</tr>
<tr>
<td>5</td>
<td>0.77</td>
<td>0.46</td>
</tr>
<tr>
<td>10</td>
<td>0.81</td>
<td>0.27</td>
</tr>
<tr>
<td>20</td>
<td>0.88</td>
<td>0.17</td>
</tr>
</tbody>
</table>

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.
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 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 7.** Experimental results for automatic probing.

<table>
<thead>
<tr>
<th></th>
<th>Femur</th>
<th>Pelvis</th>
<th>Bunny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>RMS (mm)</td>
<td>Time (s)</td>
<td>RMS (mm)</td>
</tr>
<tr>
<td>dSPNR</td>
<td>1.5</td>
<td>2.17</td>
<td>1.56</td>
</tr>
<tr>
<td>pSPNR</td>
<td>3.38</td>
<td>2.21</td>
<td>3.01</td>
</tr>
<tr>
<td>ICP</td>
<td>0.05</td>
<td>4.41</td>
<td>0.04</td>
</tr>
<tr>
<td>CPD</td>
<td>0.06</td>
<td>16.27</td>
<td>0.04</td>
</tr>
<tr>
<td>GoICP</td>
<td>37.49</td>
<td>4.41</td>
<td>40.13</td>
</tr>
</tbody>
</table>

---

**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 8.** Experimental results for telemanipulated probing.

<table>
<thead>
<tr>
<th></th>
<th>Femur</th>
<th>Pelvis</th>
<th>Bunny</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>RMS (mm)</td>
<td>Time (s)</td>
<td>RMS (mm)</td>
</tr>
<tr>
<td>dSPNR</td>
<td>1.5</td>
<td>2.17</td>
<td>1.56</td>
</tr>
<tr>
<td>pSPNR</td>
<td>3.38</td>
<td>2.21</td>
<td>3.01</td>
</tr>
<tr>
<td>ICP</td>
<td>0.05</td>
<td>4.41</td>
<td>0.04</td>
</tr>
<tr>
<td>CPD</td>
<td>0.06</td>
<td>16.27</td>
<td>0.04</td>
</tr>
<tr>
<td>GoICP</td>
<td>37.49</td>
<td>4.41</td>
<td>40.13</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>Number of normals</th>
<th>Number of points</th>
<th>dSPNR Time (s)</th>
<th>RMS (mm)</th>
<th>pSPNR Time (s)</th>
<th>RMS (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0</td>
<td>0.27</td>
<td>3.86</td>
<td>1.43</td>
<td>1.59</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>4.30</td>
<td>3.33</td>
<td>2.00</td>
<td>1.23</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>0.39</td>
<td>4.26</td>
<td>1.69</td>
<td>1.51</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>6.38</td>
<td>2.96</td>
<td>1.94</td>
<td>1.13</td>
</tr>
</tbody>
</table>

**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.

**Table 9.** Experimental results for telemanipulated probing.

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>RMS (mm)</td>
<td>Time (s)</td>
<td>RMS (mm)</td>
</tr>
<tr>
<td>dSPNR</td>
<td>2.26</td>
<td>3.95</td>
<td>2.14</td>
<td>3.82</td>
</tr>
<tr>
<td>pSPNR</td>
<td>3.02</td>
<td>36.40</td>
<td>4.34</td>
<td>3.93</td>
</tr>
<tr>
<td>ICP</td>
<td>0.05</td>
<td>3.83</td>
<td>0.05</td>
<td>5.09</td>
</tr>
<tr>
<td>CPD</td>
<td>0.99</td>
<td>13.58</td>
<td>1.21</td>
<td>14.14</td>
</tr>
<tr>
<td>GoICP</td>
<td>19.96</td>
<td>3.83</td>
<td>19.62</td>
<td>4.82</td>
</tr>
</tbody>
</table>

**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.
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

References


