Chapter 1
Sparse Point Registration

Rangaprasad Arun Srivatsan[1], Prasad Vagdargi[1] and Howie Choset[1*†]

Abstract This work introduces a *Sparse Point Registration* (SPR) method for performing robust registration given the geometric model of the object and few sparse point-measurements (< 20) of the object's surface. Such a method is of critical importance in applications such as probing-based surgical registration, manipulation, etc. Our approach for SPR is iterative and in each iteration, the current best pose estimate is perturbed to generate several poses. Among the generated poses, the best pose as evaluated by an inexpensive cost function is used to estimate the locally optimum registration. This process is repeated, until the pose converges within a tolerance bound. Two variants of the SPR are developed: deterministic (dSPR) and probabilistic (pSPR). Compared to the pSPR, the dSPR is faster in converging to the local optimum, and requires fewer parameters to be tuned. On the other hand, the pSPR provides uncertainty information in addition to the registration estimate. Both the approaches were evaluated using various standard data sets and the compared to results obtained using state-of-the-art methods. Upon comparison with other methods, both dSPR and pSPR were found to be robust to initial pose errors as well as noise in measurements. The effectiveness of the approach is also demonstrated with an application of robot-probing based registration.

1.1 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

* *This work was was supported by NRI Large grant IIS-1426655

† [1]R. A. Srivatsan, P. Vagdargi and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA (rarunsrivatsan@,prasadv@,choset@cs.)cmu.edu

point measurements of the object's surface, needs to be estimated. In most applications, the point clouds obtained from sensors such as LIDAR, Kinect, feature rich stereo-images, etc, contain hundreds of points. Several methods have been developed to perform registration when dense point measurements are obtained [1–5]. However, these methods do not perform well when only a small number of point measurements are available, and hence in this work we develop a method for robust sparse point registration (SPR).

Sparse point registration 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 CT scan or MRI. In such applications, there is a cost associated with probing more points and the goal is to quickly and accurately register with a fewer number of measurements. Prior work either uses greater than 100 measurement points for reliable registration [4–6], or uses *a priori* knowledge of anatomical landmarks to hand-pick a small number of probing locations [7–9]. In an attempt to keep the formulation general, in this work we do not assume any prior knowledge of anatomical segments.

Our approach to SPR 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 pose 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 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 SPR (dSPR) and probabilistic SPR (pSPR). The dSPR uses iterative closest point algorithm (ICP) [1], while the pSPR uses dual quaternion filtering (DQF) [10] to estimate the pose. Note that this approach may appear similar to the stochastic ICP [11]. However, a key difference is that stochastic ICP perturbs each point by a random noise, while our approach perturbs the pose which results in each point being perturbed in a structured manner.

The dSPR is computationally faster than pSPR, but requires the perturbation related parameters to be set manually. The pSPR on the other hand uses uncertainty information to automatically set these parameters. An additional contribution of this work is extending the online formulation of DQF in [10] to process batches of point measurements.

In Sec. 1.5, the dSPR and pSPR are evaluated in simulation over a number of standard data sets and compared against popular registration methods. The results show that the SPR typically takes less than 20 points to accurately and robustly estimate the registration and is more robust to initial registration errors ($\approx 30^\circ$ orientation and $\approx$ 30mm translation). In Sec. 1.5, a general guideline is provided on how to automatically probe the object to get a good spread of sparse points.

1.2 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, Horn *et al.* developed a least squares based approach to find the spatial transformation [12]. However, in most practical applications, the correspondence is unknown. By iteratively finding the best correspondence and the optimal transformation given that correspondence, Besl *et al.* developed the iterative closest point (ICP) [1].

Several variants of the ICP have since been developed that use surface normal information [2, 13], incorporate uncertainty in measurements [3], incorporate uncertainty in finding correspondence [5], are robust to outliers [14, 15], are globally optimal [16–18], 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 [4, 19–22].

In applications where only a small number of sparse point measurements are available due to high cost of measurement, conventional registration approaches do not perform reliably. For example, in a surgical application of probing-based registration using any of the Kalman filtering variants, the computation time might be low [4, 10, 23], but the time taken to obtain greater than 100 point measurements can be high.

Prior work such as [7, 8, 24], register using a small number of carefully chosen probing points. These points are selected based on optimization of a stiffness-based quality metric [24]. 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 approach. Firstly, the workspace constraints of the robot might restrict the robot from locating all the anatomical segments. Secondly, when applied to a non-surgical domain, defining an equivalent of anatomical feature is non-trivial. There is also a vast literature on touch-based localization for robot manipulation applications. These approaches are either mean for planar localization [25–27], or time consuming [28], or assume good initial alignment [29] or assume multiple simultaneous contacts [30, 31].

The work of Ma *et al.* [9] comes closest to our approach. In [9], an unscented particle filter (UPF) is used to register an object using a small number of point measurements without relying on prior knowledge of any landmarks or segments. While the UPF uses a small number of measurements, it uses a large number of pose particles ≈ 2000 in each iteration, resulting in a large computation time.

1.3 Mathematical Background

1.3.1 Iterative Closest Point

The ICP algorithm introduced by Besl *et al.* [1] has two important steps that are repeated until convergence

1. Finding correspondences between a point set and a geometric representation of a surface.
2. Computing the transformation which minimizes the distance between corresponding points.

Algorithm 1: Iterative Closest Point Update

Input:
$A = \{a_i \in \mathbb{R}^3\}, i = 1,2,...,n$
$B = \{b_j \in \mathbb{R}^3\}, j = 1,2,...,m$
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**
3 **Correspondence**: $c_j = FindClosestPoint(T(b_j))$, $c_j \in A$
4 **Minimization**: $T = \underset{T}{\operatorname{argmin}} \sum_{j=1}^{m} \left\| c_j - T(b_j) \right\|^2$

Consider two point clouds, $A = \{a_i\}$, $a_i \in \mathbb{R}^3$, $i = 1,\ldots,n$ are n points on the geometric model of the object and $B = \{b_j\}$, $b_j \in \mathbb{R}^3$, $j = 1,\ldots,m$ are m points obtained using sensor measurements. Let $T \in SE(3)$ be the transformation that aligns A and B. The ICP algorithm is listed in Alg. 1, typically uses Horn's method [12] to perform the minimization; although there are other optimization variants as well [13].

1.3.2 Dual Quaternion Filtering

Dual quaternion filtering (DQF) is a linear Kalman filtering based approach for online pose estimation [10]. Unlike ICP, DQF is not a batch processing algorithm. It is more similar to a Kalman filter since 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 to other filtering based registration methods such as [4, 19, 20], the DQF is preferred because it is a truly linear filter without any approximations or linearizations, resulting in quick and accurate estimates. The transformation $T \in SE(3)$ 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, $d = \left((0,t)^T \odot q\right)/2$, where $\odot$ is the quaternion multiplication operator, and $t \in \mathbb{R}^3$ is the translation vector.

Srivatsan *et al.* [10] use a pair of measurements $\{(c_j,b_j),(c_{j+1},b_{j+1})\}$, to derive

$$Hq = 0, \quad H \in \mathbb{R}^{4\times 4} \tag{1.1}$$

$$H = \begin{bmatrix} 0 & -(c_j - c_{j+1} - b_j + b_{j+1})^T \\ (c_j - c_{j+1} - b_j + b_{j+1}) & (c_j - c_{j+1} + b_j - b_{j+1})^\times \end{bmatrix}, \tag{1.2}$$

$$t = \frac{c_j - c_{j+1}}{2} - Real\left(q \odot \frac{b_j - b_{j+1}}{2} \odot q^*\right), \tag{1.3}$$

where $j = 1,3,\ldots,2\alpha-1$, $\alpha = \lfloor \frac{m}{2} \rfloor$, q^* is the conjugate quaternion and $[\,]^\times$ is the operator that converts a vector to a skew-symmetric matrix.

The update equations for the Kalman filter are:

$$q_k = q_{k-1} - K_k H_k q_{k-1},$$
$$\Sigma_k^q = (I - K_k H_k)\Sigma_{k-1}^q, \quad \text{where, } K_k = \Sigma_{k-1}^q H_k^T (H_k \Sigma_{k-1}^q H_k^T + Q_k)^{-1}$$

where Σ_k^q is the uncertainty in the quaternion q_k and Q_k is the pseudo-measurement uncertainty. The translation vector t_k is obtained from q_k using Eq. 1.3. For the sake of brevity, the derivation for $Q_k = g(\Sigma_k^q, \Sigma_k^{b_j}, \Sigma_k^{c_j})$ and $\Sigma_k^t = f(\Sigma_k^q, \Sigma_k^{b_j}, \Sigma_k^{c_j})$ representing the uncertainty in the translation, are omitted here. The expressions along with their derivation can be obtained from Sec. IV.C of [10].

The standard implementation of DQF requires ≈ 100 measurements for reliable registration estimation [10]. In this work, the DQF is modified 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 Sec. 1.5, bDQF requires fewer measurements for accurate estimation compared to the standard DQF. We modify Eq. 1.1 as

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

$H_i \in \mathbb{R}^{4\times 4}$ is as defined in Eq. 1.2. The update equations of the filter remain the same. The algorithm for bDQF is shown in Alg. 2.

1.4 Modeling

Alg. 3 and Fig. 1.1 show the basic framework and the steps involved in our SPR approach.

1. The algorithm 3 is initialized using an initial pose (see Fig. 1.1(b)).
2. The current best pose is perturbed and p perturbed poses are obtained. In Fig. 1.1, $p = 3$ is chosen. The amount of perturbation is reduced over the it-

Algorithm 2: Batch Dual quaternion filtering

Input:
$A = \{a_i \in \mathbb{R}^3\}, i = 1,2,...,n$
$B = \{b_j \in \mathbb{R}^3\}, j = 1,2,...,m$
Initial transformation: $q_0 \in \mathbb{R}^4, t_0 \in \mathbb{R}^3$,
Output:
$q \in \mathbb{R}^4, t \in \mathbb{R}^3$ that aligns A and B
$\Sigma^q \in \mathbb{R}^{4\times4}, \Sigma^t \in \mathbb{R}^{3\times3}$

Initialize: $k = 1$
while *not converged* **do**
  Correspondence:
  $T_{k-1}(b_j) = t_{k-1} + Real(q_{k-1} \odot b_j \odot q^*_{k-1})$
  $c_j = FindClosestPoint(T_{k-1}(b_j))$, $c_j \in A$,
  State Update:
  $q_k = (I - K_k G_k) q_{k-1}$
  $\Sigma^q_k = (I - K_k G_k)\Sigma^q_{k-1}$
  $t_k = \frac{1}{m}\left(\sum_{j=1}^m c_j - Real\left(q_k \odot \sum_{j=1}^m b_j \odot q^*_k\right)\right)$
  $\Sigma^t_k = f(\Sigma^q_k, \Sigma^{b_j}_k, \Sigma^{c_j}_k)$
  $k = k + 1$

Algorithm 3: Sparse Point Registration

Input:
$A = \{a_i \in \mathbb{R}^3\}, i = 1,2,...,n$
$B = \{b_j \in \mathbb{R}^3\}, j = 1,2,...,m$
Initial transformation: $T_0 \in SE(3)$
Output: $T \in SE(3)$ that aligns A and B

Initialize: $T \leftarrow T_0$, $k = 0$, $\varepsilon = \inf$
while $k <$*MaxIterations OR* $\varepsilon >$*Threshold* **do**
  Perturbation: $\tilde{T}_j = T_k + \mathcal{N}(0, \Sigma_k)$, $j = 1, \ldots, p$
  Evaluate Cost Function: $O_j = \sum_{i=1}^m ||\tilde{T}_j(b_i) - c_i||$
  Locally Optimal estimate:
  $\hat{T}_k = \text{argmin}_{\tilde{T}_j} O_j$
  $T_k = ICP(A, B, \hat{T}_k)$ or $DQF(A, B, \hat{T}_k)$
  $\varepsilon_k = \sum_{i=1}^m ||T_k(b_i) - c_i||$
  if $\varepsilon_k < \varepsilon$ **then**
    $T = T_k$, $\varepsilon = \varepsilon_k$
  $k = k + 1$

erations. Refer Sec. 1.4.1 and Sec. 1.4.2 for more information on how to choose the amount of perturbation.

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: $O_j = \sum_{i=1}^{m} ||\tilde{T}_j(b_i) - c_i||,\ j = 1,\ldots,p$, where $\tilde{T}_j \in SE(3)$ and $c_i \in A$ 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 [32,33]. For example, in this work we work with a triangulated mesh model and use a quadric mesh simplification [34] as shown in Fig. 1.1(c). The more approximate the geometry, the lesser the time taken to compute O_j but greater number of iterations for the overall algorithm. In our experiment, we do not observe the level of approximation affecting the quality of the results.
4. The pose $\hat{T} = \text{argmin}_{\tilde{T}_j} O_j$, is chosen as the initial guess for a locally optimal pose estimation using ICP or bDQF. In Sec. 1.5 we discuss the advantages and limitations of using ICP over bDQF.
5. Steps 2-4 are repeated until convergence or up to a fixed number of iterations.

1.4.1 Deterministic Sparse Point Registration (dSPR)

In the dSPR, ICP as described in Sec. 1.3.1 is used to find the locally optimal pose (Step 4 of SPR). There are three tunable parameters:

1. Number of perturbations: Perturbations help the optimizer move out of a local minima. The higher the number of perturbations, the faster the convergence to the optimal estimate is. But higher perturbations also imply higher computation times. In this work we choose the number of perturbations $p = 10$.
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 10°. The initial perturbation in translation is drawn from a normal distribution with zero mean and a standard deviation of 10% of the size of the 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 RMS error between A and B is lower than a set threshold. In this work, we set the maximum number of iterations to be 30 and the RMS error threshold to be 0.5% of the size of the 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

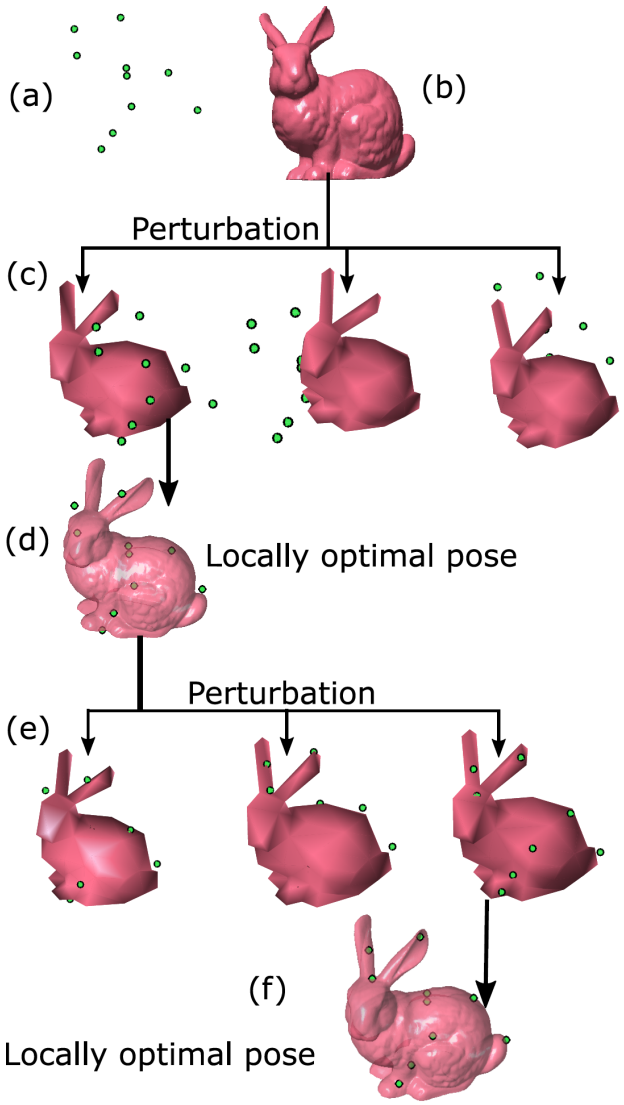

Fig. 1.1 The steps involved in two iterations of the sparse point registration algorithm. The best pose estimate in each iteration is perturbed to obtain three pose particles. (a) Point measurements (b) Geometric model of the object (c) Perturbed poses of a lower resolution geometric model. The number 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 ICP or bDQF and the original geometric model. (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 obtained and the steps are repeated until convergence.

computation time. The choices of parameters made in this work for dSPR are based on manual tuning over several standard data sets; and are not meant to exhibit any optimal behavior.

1.4.2 Probabilistic Sparse Point Registration (pSPR)

In the pSPR, bDQF as described in Sec. 1.3.2 is used to find the locally optimal pose. There are two tunable parameters:

1. Number of perturbations: Since we use a Gaussian distribution-based filter, the number of perturbations are be chosen to be equal to the number of sigma-points. In the case of bDQF, there are 15 sigma points. Unlike the dSPR, 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 be 12 and the RMS error threshold to be 0.5% of the size of the object. The number of iterations in each bDQF step is set to 50.

1.5 Simulation Experiments

We perform a number of simulation experiments on standard shape data sets to systematically study the dSPR and pSPR. The results are compared with ICP, bDQF and UPF [9]. For all the simulation experiments, the objects are scaled to fit in a cube of edge length 100mm, for a fair comparison of the registration errors [3].

1.5.1 Minimum Number of Points Required

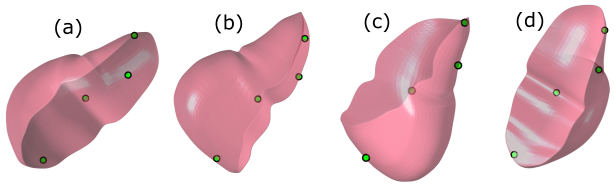

Fig. 1.2 Four different poses of the liver contain the same set of four green points.

[3] 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.

Different shapes need different number 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 [12]. If the correspondences are unknown, there may exist multiple valid solutions for the pose when only a small number of points are available (see Fig. 1.2). 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.* [7] (later extended by Ellis *et al.* [24]) comes closest to answering this question. Given the geometry of the object, Simon *et al.* find 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 mini-

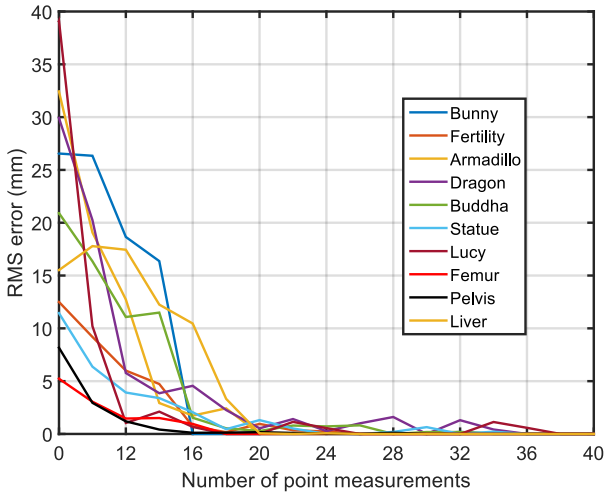

Fig. 1.3 Plot of RMS error vs number of points used for registration, when using dSPR. For each integer element on the X axis, mean error is computed over 100 experiments. Most of the shapes considered need ≈ 20 measurements for accurate registration.

mum number of point measurements required for reliable registration, we perform an experiment where p random points from the model are selected. A known transformation is applied to these p points. The applied transformation and RMS error are estimated using dSPR. 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 [35], Fertility obtained from the AIM shape repository [36], femur bone, liver obtained from https://grabcad.com and pelvis bone obtained from https://www.thingiverse.com/.

It is observed that some shapes like pelvis and Bunny require only 16 points, while others like the Dragon require 36 points. Most of the shapes need ≈ 20 points. Given a new shape, similar experiments can be run to empirically find out the minimum number of random points required for reliably registration estimate.

1.5.2 Robustness to Noise

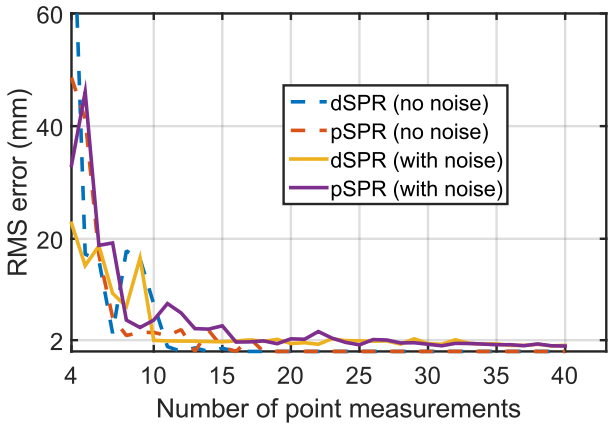

Fig. 1.4 Plot of the RMS error vs number of measurements used for dSPR and pSPR, with a without noise in the measurements. In the absence of noise, dSPR takes 12 measurements and pSPR takes 18 measurements to converge to zero RMS error. In the presence of a uniform noise of 2mm, both pSPR and dSPR converge to an RMS error of $<$ 2mm after 20 measurements.

We repeat the experiment from the previous section for a femur bone, with different amount of noise added to the points. The dashed lines in Fig. 1.4 show the mean error for dSPR and pSPR over 100 experiments versus the number of measurements used. The RMS error for both dSPR and pSPR 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. Fig. 1.4 shows that the mean error for dSPR and pSPR both converge to less than 2mm after 20 measurements. The performance of both dSPR and pSPR are very similar– with and without measurement noise. Table 1.1 shows the RMS error for dSPR and pSPR as well as the estimation time, for varying levels of noise, using 20 point measurements and 100 measurements. When using 100 measurements, as

expected, dSPR, pSPR, standard ICP, bDQF and UPF accurately estimate the registration. However, when using 20 measurements, dSPR and pSPR outperform the other methods. UPF takes the most computation time compared to all the other methods because of the need to iterate over ≈ 2000 pose particles (other researchers have also noticed the high computation time of UPF [4]).

In this work, we do not present comparisons with other popular registration methods such as generalized ICP [3], UKF-based registration [4], DQF [10], iterative most likely point [5], etc. as those methods are not designed to work with less than 100 measurements and hence the comparison would be unfair.

Table 1.1 Femur bone: Registration in the presence of noise

	No noise		2mm noise		5mm noise	
20 points	RMS (mm)	Time (sec)	RMS (mm)	Time (sec)	RMS (mm)	Time (sec)
dSPR	0	0.08	1.08	0.44	1.79	0.46
pSPR	0	0.07	1.63	1.009	1.66	0.99
ICP	4.72	0.01	2.92	0.01	6.69	0.01
bDQF	2.84	0.05	2.19	0.09	7.83	0.09
UPF	7.13	161.77	22.12	159.83	24.23	242.38
100 points						
dSPR	0	0.03	0.59	0.51	1.12	0.53
pSPR	0	0.13	0.64	2.53	1.07	3.10
ICP	0	0.01	0.72	0.01	0.93	0.01
bDQF	0	0.08	0.38	0.19	1.55	0.24
UPF	4.82	1891.1	8.06	1843.3	14.29	1594.0

Even though the RMS error is similar, the time taken by pSPR is greater than dSPR. This is because each ICP evaluation in dSPR internally takes ≈ 20 iterations, while each bDQF evaluation in pSPR internally takes ≈ 50 iterations for convergence to a local minima. The benefit of using pSPR however, lies in the fewer parameters that require tuning.

1.5.3 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 that have been developed in this work. For both the strategies, first find the location and dimensions of a cuboid in the workspace of the robot, within which the object lies. The object can be probed from 5 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:

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 Fig. 1.5(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 $< 30\%$ of the largest face), then the previous strategy would result in most of the probed points lying on the face with the largest area. So 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 Fig. 1.5(b)).

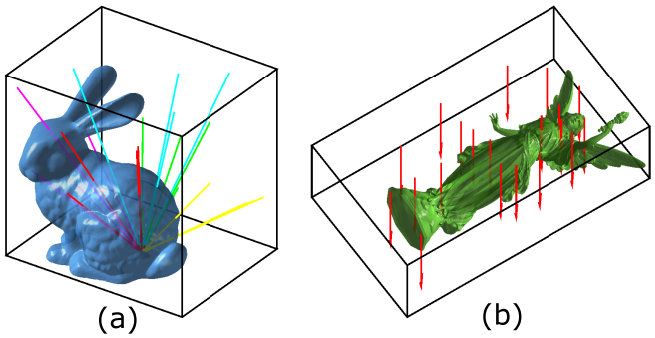

Fig. 1.5 A cuboid is selected in the workspace of the robot that conservatively estimates the location of the object. (a) Different probing paths for the robot are selected such that the probed points are spread across the surface of the object. The colors of the path show the face of the cuboid that 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.

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 [30], or manually by a teleoperator [37].

1.6 Robot Experiments

In order to test our approach to SPR with real data, an experimental setup as shown in Fig. 1.6(a) is used. The setup consists of a 6-DOF robot Foxbot® equipped with an ATI Nano17 force sensor at the end-effector. The object of interest is clamped in

front of the robot and is probed using the strategies described in Sec. 1.5.3. The objects chosen for this experiment are: Femur bone, pelvis bone and Stanford bunny. The largest dimension for each of these objects is 100mm. Using the information obtained from Fig. 1.3, we collect 18, 20 and 20 points respectively for the femur, pelvis and bunny respectively. The blue circles in Fig. 1.6(b)-(d) show the initial

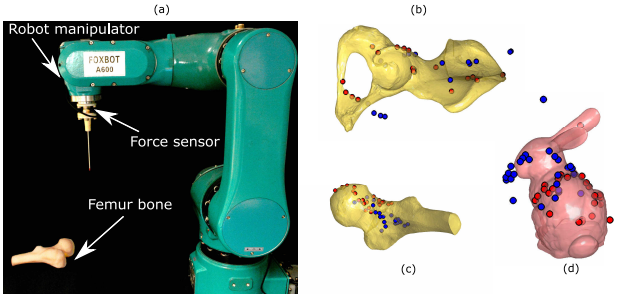

Fig. 1.6 (a) Experimental set up consists of a robot manipulator equipped with a force sensor. The object that is to be registered is clamped and held in place. (b)-(d) Blue circles represent the initial location of the point measurements and red circles represent the registered location of the points. Pelvis bone, femur bone and bunny are probed at 18, 20 and 20 points respectively.

guess for the location of the point measurements and the red circles show the location as estimated by dSPR. Note that the estimated location of the points lie on the model of the objects. The RMS error for the location estimated by dSPR and pSPR are shown in Table 1.2. For 20 point measurements, the RMS error for bunny is higher than the other two models as is consistent from our simulation experiments (see fig. 1.3).

Table 1.2 Experimental results

	Pelvis		Femur		Bunny	
	RMS (mm)	Time (sec)	RMS (mm)	Time (sec)	RMS (mm)	Time (sec)
dSPR	2.17	1.5	1.38	1.56	5.00	1.76
pSPR	2.21	3.38	2.33	3.01	4.91	3.95

1.7 Conclusion

In this work, a sparse point registration (SPR) method for robust registration using a small number of sparse point measurements was developed. The approach can be implemented in a deterministic manner (dSPR) or a probabilistic manner (pSPR).

The dSPR is faster but has more parameters to tune than pSPR. pSPR has the added advantage over dSPR of providing the uncertainty in the registration estimate. We also provide guidelines for choosing the tuning parameters for each algorithm. Another contribution of this work is the development of a batch processing variant of the dual quaternion filter [10], which is used in the pSPR.

Through simulations and robot experiments, both dSPR and pSPR are found to be robust and accurate compared to state-of-the-art methods. Even in the presence of noise, our approach accurately estimates the registration compared to popular deterministic and probabilistic approaches for registration. The computation time was $\approx$ 1s for dSPR and $\approx$ 3s for pSPR. A C++ implementation would greatly reduce the computation time. Through a number of simulations, it is empirically found that most shapes require $\approx$ 20 points 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.

In the future, we plan to extend the formulation to flexible objects that can deform upon contact. Prior work such as [23] can be used to estimate the local deformation introduced by forceful contact. Future work will also explore using surface-normal in addition to point measurements for registration.

References

1. P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.
2. S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings of 3rd International Conference on 3-D Digital Imaging and Modeling*. IEEE, 2001, pp. 145–152.
3. A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: Science and Systems*, vol. 2, no. 4, 2009.
4. M. H. Moghari and P. Abolmaesumi, "Point-based rigid-body registration using an unscented Kalman filter," *IEEE Transactions on Medical Imaging*, vol. 26, no. 12, pp. 1708–1728, 2007.
5. S. D. Billings, E. M. Boctor, and R. H. Taylor, "Iterative most-likely point registration (IMLP): A robust algorithm for computing optimal shape alignment," *PloS one*, vol. 10, no. 3, 2015.
6. M. A. Audette, F. P. Ferrie, and T. M. Peters, "An algorithmic overview of surface registration techniques for medical imaging," *Medical image analysis*, vol. 4, no. 3, pp. 201–217, 2000.
7. D. A. Simon, M. Hebert, and T. Kanade, "Techniques for fast and accurate intrasurgical registration," *Journal of image guided surgery*, vol. 1, no. 1, pp. 17–29, 1995.
8. B. Ma and R. E. Ellis, "Robust registration for computer-integrated orthopedic surgery: laboratory validation and clinical experience," *Medical image analysis*, vol. 7, pp. 237–250, 2003.
9. ——, "Surface-based registration with a particle filter," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2004, pp. 566–573.
10. R. A. Srivatsan, G. T. Rosen, F. D. Naina, and H. Choset, "Estimating SE(3) elements using a dual quaternion based linear Kalman filter," in *Robotics : Science and Systems*, 2016.
11. G. P. Penney, P. J. Edwards, A. P. King, J. M. Blackall, P. G. Batchelor, and D. J. Hawkes, "A stochastic iterative closest point algorithm (stochastICP)," in *MICCAI*, 2001, pp. 762–769.
12. B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.
13. G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: a survey from rigid to nonrigid," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 7, pp. 1199–1217, 2013.

14. Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *European conference on computer vision*. Springer, 2004, pp. 558–569.
15. J. M. Phillips, R. Liu, and C. Tomasi, "Outlier robust ICP for minimizing fractional RMSD," in *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*. IEEE, 2007, pp. 427–434.
16. N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on geometry processing*, vol. 2, no. 3, 2005, p. 5.
17. A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *CVPR*, vol. 1. IEEE, 2006, pp. 1297–1304.
18. J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *IEEE International Conference on Computer Vision*, 2013, pp. 1457–1464.
19. X. Pennec and J.-P. Thirion, "A framework for uncertainty and validation of 3-D registration methods based on points and frames," *International Journal of Computer Vision*, vol. 25, no. 3, pp. 203–229, 1997.
20. S. Hauberg, F. Lauze, and K. S. Pedersen, "Unscented Kalman filtering on Riemannian manifolds," *Journal of Mathematical Imaging and Vision*, vol. 46, no. 1, pp. 103–120, 2013.
21. R. A. Srivatsan, M. Xu, N. Zevallos, and H. Choset, "Bingham distribution-based linear filter for online pose estimation," in *Robotics: Science and Systems*, 2017.
22. R. A. Srivatsan and H. Choset, "Multiple Start Branch and Prune Filtering Algorithm for Nonconvex Optimization," in *WAFR*, 2016.
23. R. A. Srivatsan, E. Ayvali, L. Wang, R. Roy, N. Simaan, and H. Choset, "Complementary Model Update: A Method for Simultaneous Registration and Stiffness Mapping in Flexible Environments," in *International Conference on Robotics and Automation*, 2016, pp. 924–930.
24. B. Ma and R. E. Ellis, "Spatial-stiffness analysis of surface-based registration," in *MICCAI*, 2004, pp. 623–630.
25. K. Gadeyne and H. Bruyninckx, "Markov techniques for object localization with force-controlled robots," in *Intl Conf. on Advanced Robotics*, 2001.
26. S. R. Chhatpar and M. S. Branicky, "Particle filtering for localization in robotic assemblies with position uncertainty," in *IROS*, 2005, pp. 3610–3617.
27. K. Hsiao and L. P. Kaelbling, "Task-driven tactile exploration," in *Robotics: Science and Systems*, 2010.
28. A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.
29. B. Saund, S. Chen, and R. Simmons, "Touch based localization of parts for high precision manufacturing," in *International Conference on Robotics and Automation*. IEEE, 2017.
30. S. Javdani, M. Klingensmith, J. A. Bagnell, N. S. Pollard, and S. S. Srinivasa, "Efficient touch based localization through submodularity," in *International Conference on Robotics and Automation*. IEEE, 2013, pp. 1828–1835.
31. P. Hebert, T. Howard, N. Hudson, J. Ma, and J. W. Burdick, "The next best touch for model-based localization," in *ICRA*, 2013, pp. 99–106.
32. M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Conference on Visualization'02*, 2002, pp. 163–170.
33. P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, 1998.
34. M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 209–216.
35. G. Turk and M. Levoy, "The Stanford 3D Scanning Repository," *Stanford University Computer Graphics Laboratory http://graphics.stanford.edu/data/3Dscanrep*.
36. AIM@SHAPE model repository, "Fertility point cloud scan," *http://visionair.ge.imati.cnr.it/ontologies/shapes/releases.jsp*.
37. R. A. Srivatsan, P. Vagdargi, N. Zevallos, and H. Choset, "Multimodal registration using stereo imaging and contact sensing," in *Robotics: Science and Systems, Workshop on 'Revisiting Contact - Turning a problem into a solution'*, 2017.