Probabilistic Approaches for Pose Estimation

Arun Srivatsan Rangaprasad

May 2018

CMU-RI-TR-18-24

The Robotics Institute

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

Thesis Committee

Howie Choset (chair)

Michael Kaess

Simon Lucey

Russell H. Taylor, JHU

Nabil Simaan, VU

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy

Abstract

Virtually all robotics and computer vision applications require some form of pose estimation; such as registration, structure from motion, sensor calibration, etc. This problem is challenging because it is highly nonlinear and nonconvex. A fundamental contribution of this thesis is the development of fast and accurate pose estimation by formulating in a parameter space where the problem is *truly* linear and thus globally optimal solutions can be guaranteed. It should be stressed that the approaches developed in this thesis are indeed inherently linear, as opposed to using linearization or other approximations, which are known to be computationally expensive and highly sensitive to initial estimation error.

This thesis will demonstrate that the choice of probability distribution significantly impacts performance of the estimator. The distribution must respect the underlying structure of the parameter space to ensure any optimization, based on such a distribution, produces a globally optimal estimate, despite the inherent nonconvexity of the parameter space.

Furthermore, in applications such as registration and 3D reconstruction, the correspondence between the measurements and the geometric model is typically unknown. In this thesis we develop probabilistic methods to deal with cases of unknown correspondence. While the resultant formulation does not guarantee global optimality, it increases the basin of convergence.

Another contribution of this thesis is the development of a generalized framework for probabilistic point registration. By setting functional parameters such as sensor noise and prior uncertainties appropriately, our framework captures many prior registration methods. Additionally our framework is also capable of predicting scope for improvement in the existing algorithms, which are then verified experimentally.

To tie the ideas together, we present results in the context of surgical robotics - in particular we demonstrate a surgical system that is capable of performing real-time tumor localization, hand-eye calibration, registration of preoperative models to the anatomy, and augmented reality.

Acknowledgments

Given that this thesis is more than 300 pages long, it should not come as a surprise that the acknowledgment section is a four pages long. There are three parts to the acknowledgment – (1) dedication to my mentors, (2) colleagues and coauthors of the papers I have written, and (3) family and friends.

Mentors

I am thankful to my adviser, Dr. Howie Choset, who has inspired me and pushed me to work harder than I ordinarily would have. He made me realize that searching for "intellectual nuggets" is a lot more valuable than finding a topic to write a one-off conference paper. He has helped me strike a fine balance between theoretical work and practical experiments. He has also spent countless days trying to help me improve my writing skills and presentation skills, without which this dissertation would have been impossible. In addition to guidance with research, he has also helped me on several occasions with my personal problems, has always believed in me and supported me when it mattered the most, for which I will forever be indebted to him. I wish I end up being at least half as successful a researcher, as him.

I am also very thankful to Dr. Nabil Simaan who has always treated me like his own PhD student. He has always taken a personal interest in my growth and provided me with constructive feedback to improve my research. I have enjoyed and greatly benefited from collaborating with his group at Vanderbilt university. His inputs have directly resulted in the contributions of Chapter 3, Chapter 8, Chapter 4 and Chapter 10. I am amazed at how Dr. Simaan can remember the details of every research paper he has ever read and talk about them on demand. That is a super power I wish I had.

My sincerest thanks to Dr. Russ Taylor, for his unique insights and consistent guidance throughout my research. He was the one who suggested the name complementary model

update (CMU), for the method developed in Chapter 3. He has been a great critique of my work and has helped me be clear and concise about my contributions and differences from the prior approaches.

I would like to thank Dr. Matthew Travers for introducing me to the paper on banana distribution [1], which let to the contributions of Chapter 5 as well as the first paper that I published in my PhD. I would also thank Dr. Michael Kaess, Dr. Simon Lucey and Dr. George Kantor for providing me with valuable feedback from time to time and serving on my committee. I would also like to thank our collaborators Dr. Peter Kazanzides and Dr. John Galeotti for their insights and support.

I am thankful to Elif Ayvali and Siddharth Sanan who were postdoctoral researchers in the biorobotics lab and helped me a lot during my PhD. Elif in particular was instrumental in teaching me how to patiently pursue research and take positives from every failure and move on.

I am also extremely grateful to Dr. Sandipan Bandopadhyay, who was my master's adviser. Without his guidance and inputs, I would have never gotten into robotics, leave alone a PhD.

Coauthors and colleagues

I am very thankful to have worked with some great people over the course of the PhD. Table 1 shows the list of people that I had the good fortune to work with on various projects (Table does not include mentors who have been acknowledged in the previous section).

Table 1: Chapters vs people Arun is thankful for

Chapter	People
3	Long Wang, Rajarshi Roy
4	Gillian Rosen, Feroze Naina, Nicolas Zevallos,
5	Stephen Tully, Michael Schwerin, David Kohanbash
6	Nicolas Zevallos, Mengyun Xu (Olivia), Feroze Naina
8	Prasad Vagdargi, Brad Saund, James Picard
10	Nicolas Zevallos, Lu Li, Hadi Salman, Bocheng Yu, Rashid Yasin, Long Wang, Saumya Saxena, Kartik Patath, Tejas Zodage, Jianing Qian

Family and friends

Senior students Chaohui, Dave, Matt Tesch, Glenn and Tony have all been great role models for me to look up to, and have over the time given me some great tips to deal with my roadblocks, paper rejections, and home sickness. I am happy to have worked with Steve Tully, Alex Ansari, Hadi, Simon, Puneet, Jaskaran, Guillaume, Julian, Richard, Steve and Saumya. I will miss the innumerable discussions, fun trips, movies and long rants that we have had. I must also acknowledge the hard work of Olivia, Gillian, Nate Appelson, Feroze, Prasad, Kartik Patath and Tejas, which resulted in many of the experimental results in this thesis. Without Lu Li, Jim Picard and David Kohanbash, a lot of the electronics, mechanical fabrication and systems work would not have been possible. Finally, I feel Nico Zevallos must be given at the very least, a masters degree, for the amount of work he has done to support my research. Nico has been a coauthor on at least seven papers with me and without his systems development work, my thesis would not look half as good as it currently does. I hope after this PhD, Nico and I would be able to find some time to pursue a few art projects that we have been thinking about including "Painting miniature sculptures of baby Jesus using the daVinci robot" (concepts from Chapter 3, Chapter 6 and Chapter 10 should be useful in this task).

I have gained many important insights from friends and colleagues at CMU. I would specially like to thank Abhijeet who was also my officemate for five years. His productivity was reduced by half due to my constant chatter and discussion about research as well as non-research topics ranging from 'spiritual gurus' to 'deep learning'. I am also grateful to have great support from Jayanth, VenkatG, Tejas, Nate Wood, Humphrey, Aayush Bansal, Fahad, and Sid; who in spite of their own PhD/masters woes, always took out time to help me with my issues.

Special thanks to the excellent support staff at CMU, in particular, Peggy, who is now almost a part of my family. Suzanne has always been a harbinger of good news, starting with the announcement of my admission to RI. I will miss going to Udipi with her. I am also lucky to have met Jean who is the kindest (and also most hard working) human in RI. I must also thank Sumitra and Sudhir for taking a special interest in my well being.

Outside CMU, I must thank Long and his wife Rachel for being excellent hosts every time we visited Nashville. I also wish to thank Preetham, and Anton from JHU for the

several dVRK debugging sessions. I am very thankful to the National Robotics Initiative (NRI) for not only supporting our research, but also providing opportunities to make such great friends for life. I am also thankful to the Center for Machine Learning and Health (CMLH), for providing a fellowship to support my research in the last year of PhD.

Cricket has been an important part of my life, and thanks to Ahsan, Sahil, Prajna and Sameen bhai I have managed to play competitive cricket in Pittsburgh, which has been a very pleasant distraction from research. I thank all my teammates from CMU Tartans and Kent cricket club as well as many players I have played with in the Pittsburgh cricket association and some from the United States cricket team.

I have met some people serendipitously in Pittsburgh, who have now become an important part of my life. Thank you Meghana, Bhavana, Anjali, Aimon, Vivek, Karthik (SPEVK), Kalyani, Vamshi, Shinjini, Anusha, KJ, Ashwati, Divya and Nisarg for being great pillars of support and for always caring for me and treating me like your brother. There are several ups and downs when pursuing a PhD, and Ruta has always stood by me and supported me. Without her support and care, this thesis would have been impossible. There are no words that exist to sufficiently acknowledge her inputs in keeping me sane to pursue research in the last 5 years.

I am happy that some of my friends from India, have taken the time to come visit me in Pittsburgh during my PhD. Special thanks to Tarun, Kartik, Swarna, Bhavin, Deepak and Kedar. I am also thankful to Darshan, for hosting me in UK and always checking on me and lifting my mood. I am blessed to have a sister in Ketaki, who has never forgotten to send me rakhi and gifts during my birthdays. I am also thankful to Ganesh, Vani, Swapnil, Vaibhav, Sriramcharan and Hem.

Finally, I wish to apologize to my parents, grandparents and my sister, Varsha, for staying so far away from them for nearly half a decade. I have missed them every day and its their constant belief in my abilities and support, that has given me the strength to finish this PhD. Explaining my research to my grandmother has helped me tremendously in organizing my thoughts and articulating my research during various presentations. I feel lucky to have the best sister in the world, who instantly cheers me up and makes me feel like a hero. I dedicate Fig. 7-6 to my mother. My father has read every research paper I have written and provided me with feedback in the best way he could. I am so indebted to my parents for everything they have provided me with and I hope this thesis makes them

proud.

After writing such a long acknowledgment, if I have missed names of anyone, it is only because of lack of sleep. I consider myself lucky to have known so many people who care about me and have taken an active interest in my PhD and my well being in general.

Dedicated to Chanakya

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Several applications in robotics require estimation of pose (translation and orientation) between two references frames of interest; applications include medical image registration [5], manipulation [9], hand-eye calibration [10] and navigation [9]. Depending on the nature of the sensor measurements, frequency of receiving the measurements, knowledge of data association between the measurement modalities and computational constraints imposed by the application, pose estimation offers different challenges in different applications. As a result, a variety of approaches have been developed in literature to cater to the unique challenges offered by different applications [11, 12, 6, 10, 9, 13].

This thesis uses a probabilistic reasoning to estimate pose as opposed to deterministic methods. We believe that a probabilistic approach has the potential to reason better about uncertainties that exist in the measurements, and our prior understanding of the problem. Furthermore, the uncertainty in the state can guide the search for the globally optimal pose. The thesis provides improvement to the state-of-the-art by deriving linear models for probabilistic pose estimation and by using the appropriate parameter space and probability distributions that respect the underlying structure of the space. This results in fast, accurate and globally optimal pose estimates for a variety of pose estimation problems when data association is known and locally optimal estimates when data association is unknown. This thesis also develops a framework which captures several existing works in literature, as its special case. Thus providing

the theoretical foundation required to understand the existing works in literature and help researchers decide which method would be most suitable for their application.

1.1 Related Work

1.1.1 Pose Estimation/ Registration

Much of the early literature deals with collecting all sensor measurements and processing them offline in a batch to estimate the pose. Registration is an important pose estimation problem that is frequently encountered in robotic applications, such as computer vision [14], localization and mapping [15], surgical guidance [16], etc. Registration is the process of finding a spatial transformation that aligns a point set with a geometric model (or another point set). Horn *et. al.* [4] and Arun *et. al.* [17] developed a least squares implementation for registration with known point correspondence. Besl *et. al.* [11] introduced the ICP, which extends Horn's methods for unknown point correspondence by iteratively estimating point correspondence and performing least squares optimization. Several variants of the ICP have been developed [12, 18, 19, 20]. Estepar *et. al.* [21], Segal *et. al.* [22] and Billings *et. al.* [23] further generalized the ICP by incorporating sensor uncertainties.

In applications such as touch-based localization [24], hip-replacement surgery [25], finger-print matching [26], 3D reconstruction using range scans [27], etc., normals are available in addition to point measurements. Several registration approaches have been developed that use orientation data in addition to point data for registration. The approach of Pulli *et. al.* [28] uses surface-normal information to filter out measurements during the correspondence stage. Munch *et. al.* [29] use point and surface-normal measurements in both the correspondence and minimization step. Billings *et. al.* [25] have recently developed iterative most likely oriented point (IMLOP), a probabilistic framework to estimate pose using surface-normal and position measurements, while incorporating measurement uncertainty in both the correspondence and minimization step.

In addition to point and surface-normal measurements there have also been several approaches developed for pose estimation using pose measurements for applications such as hand-eye calibration. Hand-eye calibration has been posed as a least squares estimation problem assuming complete knowledge of measurement correspondence [30, 31, 32]. More recently, Ackerman *et. al.* [33] estimate the pose with unknown correspondence.

With the advent for large volumes of easily shareable labeled-datasets, learning-based approaches have gained popularity recently. Supervised learning approaches such as [34, 35], provide good initial pose estimates and rely on local optimizers such an ICP for refined estimates. These methods are however not very accurate and generalize poorly to unseen object instances [36]. Recent work has attempted to leverage the power of deep convolutional neural network (CNN) to overcome these limitations. The works of [37, 38, 39, 40, 41], in particular directly regress over the Euler angles of object orientations from cropped object images. On the other hand, in applications such as robotic manipulation, 6-DOF pose is often decoupled into rotation and translation components and each is inferred independently [37, 42, 43, 44, 45, 46].

1.1.2 Sequential Pose Estimation

In some applications of pose estimation such as structure from motion [15], hand-eye calibration [31], localization [47, 48], etc. a stream of measurements is available and it is desirable to have a sequential estimator to process the measurements online. While the batch processing methods described earlier can be applied repeatedly on incremental data to deal with sequential measurements, such an approach has been observed to often perform poorly [5]. On the other hand, filtering-based methods that model pose as a distribution and perform a maximum *a posteriori* (MAP) estimate, are better at handling sequential measurements. The uncertainty in the state variables is often modeled using PDF and the parameters of the PDF are updated after each measurement. In contrast to batch estimation methods, where there is no indication of when to stop collecting measurements, convergence of the state esti-

mate and decrease in the state uncertainty provides clear indication of when to stop collecting measurements

Several sequential estimation methods are based on Kalman filters, which model the states and measurements using Gaussian distributions ([6, 5, 10, 49, 50]). Kalman filters by construction provide optimal state estimates when the process and measurement models are linear and the states and measurements are Gaussian distributed ([51]). Pose estimation, however, is inherently a non-linear problem, and hence linear Kalman filters produce poor estimates ([6, 49]). Several variants of the Kalman filter have been introduced to handle the non-linearity. EKF-based filters perform first-order linear approximations of the non-linear models [6]. UKF-based methods can be shown to perform a second-order approximation of the nonlinear model by evaluating and propagating the motion models at multiple deterministically sampled points $SE(3)$ ([5, 49]).

There has been some recent work in robotics towards the use of alternative distributions to model the noise on rotations for pose estimation problems. For example, Langevin distributions have been used for pose estimation [52, 53]. Gilitschenski *et. al.* [54] have recently developed a Bingham distribution-based recursive filtering approach for orientation estimation. Glover *et. al.* [13] use Bingham distribution to describe the orientation features, while Gilitschenski *et. al.* [55] use this distribution for planar pose estimation. In a more recent work, [56] use a constrained Kalman filter with dual quaternions as state vector.

The use of Bingham distribution to model uncertainties in rotation parameters is a very valuable tool that has been largely under-utilized by the robotics community, as also noted by [13]. One of the important reasons for this, is the difficulty in computing the normalization constant as well as performing expensive convolution operation over the distributions ([57]).

1.1.3 Parameter Space for Pose Estimation

There are several ways to parameterize pose and each choice of parameterization comes with its own advantages [58, 59]. Faugeras *et. al.* [60] and Horn [4] use

quaternions and Cartesian coordinates to parameterize the pose. Arun *et. al.* [17] use an orthogonal matrix and Cartesian coordinates, while Walker *et. al.* [61] use dual quaternions to parameterize the pose. Moghari *et. al.* [5] use Euler angles and Cartesian coordinates to parameterize the pose and a UKF to estimate the pose. Hauberg *et. al.* [49] confine the state variables over a known Riemannian manifold and use a UKF to estimate the pose. Quaternions are used to parametrize $SO(3)$ and the state is estimated using an EKF in [62] and UKF in [63]. An iterative extended Kalman filter (IEKF) with dual quaternions to parameterize the pose has been used in [64]. Fan *et. al.* [56] use a constrained Kalman filter with dual quaternions as state vector. More recently Brossard *et. al.* [65] use a UKF on Lie groups for visual inertial odometry.

1.1.4 Dealing with Nonconvexity in Pose Estimation

Pose estimation is inherently a nonconvex optimization problem. The nonconvexity is further pronounced when the correspondence between the measurements in the two frames is unknown. The situation further worsens when there are only a small number of sparse measurements available to estimate the pose and the object is symmetric. There are broadly two approaches to deal with the issue of nonconvexity– i) use global optimization approaches to find the best pose, ii) use heuristics to find 'good' correspondence.

Fitzgibbon *et. al.* [66] developed a Levenberg-Marquardt-based approach to simultaneously optimize for both correspondence and pose parameters. Yang *et. al.* [18] developed globally optimal ICP (Go-ICP), a branch and bound-based optimization approach to obtain globally optimal pose. More recently convex relaxation has been used for global pose estimation using Riemannian optimization [67], semi-definite programming [68, 69] and mixed integer programming [20]. A major drawback of the above methods is the large computation time, rendering them unsuitable for real time applications. Further, none of the above methods consider uncertainty in measurements or in pose in their formulation.

Using heuristics to find correspondence can improve the computationally speed of

the registration approaches. However, heuristics are application dependent and what works for one application may not for another. For instance, in probing-based registration for surgical applications, anatomical segments and features can be easily identified by visual inspection. Choosing these features or points within these segments can greatly help improve the correspondence as shown in [70, 71, 72]. In applications where structured light (RGBD sensors) or range scanners are used, the point measurements obtained are surprisingly dense, but they may be only partially descriptive of the object to be registered. A variety of geometric features can be constructed from this data and used for finding correspondence. The works of Gelfand *et. al.* [27] use scale invariant curvature features, Glover *et. al.* [13] use oriented features and Bingham Procrustean alignment, Makadia *et. al.* [73] use extended Gaussian images, Rusu *et. al.* [74] use fast point feature histograms, and Godin *et. al.* [75] use color intensity information for correspondence matching. When dealing with applications where volumetric data is available, curve-skeletons [76] and heat kernel signature [77] can be used to obtain a good initial estimate for the pose.

1.2 Story of This Thesis

The work in this thesis began as an interest in the problem of improving situational awareness for surgeons performing minimally invasive surgery. The idea was to fuse preoperative information such as CT scan or magnetic resonance imaging (MRI) with information obtained during the surgery from stereo cameras, contact force measurements, etc. to provide improved situational awareness to the surgeon. As a first step towards this, we decided to perform force regulated probing of the tissue surface to detect tumors and other stiff inclusions while simultaneously using position information to localize the tool tip onto a geometric model of the anatomy obtained from preoperative imaging. Our group named this problem, SCAR [78]. Our group had earlier developed a filtering-based approach for SCAR that updates the state using a stream of force and position measurements [78]. However, we made an observation that every time the state was updated, there was an ambiguity in the registration

along the normal direction that the anatomy is probed along. This ambiguity in turn resulted in incorrect stiffness measurements. More details on this can be obtained from Chapter 3. Our solution to resolve this ambiguity was to update the state based on a mini-batch of "compatible" measurements. From each batch of compatible measurements, a pseudo-measurement was created and used to update the state. This helped resolve the ambiguity and produce accurate estimates of both pose and stiffness distribution. This experience provided us a new way to think about using Kalman filters, where one could use a mini-batch of measurements selected based on a criteria; instead of the vanilla implementation that updates using every measurement sequentially.

We then applied this experience to develop an improved filter for pose estimation. We select pairs of point measurements (or mini-batches of measurements) that are spatially spaced apart. We then formulate a measurement model using dual quaternions to parameterize the pose. The resultant update model decoupled orientation from translation estimation and is linear in both orientation and translation. This allowed us to use a linear Kalman filter to estimate the pose more accurately than earlier approaches that required linearization or other forms of approximation. Subsequently we modified the filter to process surface normal and pose measurements as well. One aspect of the filter that was not satisfactory was the use of Gaussian distribution to model uncertainty in unit quaternions. Gaussian distributions are unimodal, while unit quaternions $\widetilde{\boldsymbol{q}}$ and $-\widetilde{\boldsymbol{q}}$ are equivalent and uncertainty in them is best described by a bi-modal distribution. Further, the Kalman filter ensured that the mean of the distribution lied on a unit-hypersphere (this constraint is imposed by the unit norm condition of the rotation quaternion). But the distribution in general did not respect the constraints of the parameter space. More details on this can be obtained from Chapter 2 and Chapter 4.

While searching for a better parameterization and choice of probability distribution for pose estimation, we came across the work of Long *et. al.* [1]. Long *et. al.* showed that stochastic systems evolving on $SE(3)$ follow a banana-like distribution in the space of Euler angles and Cartesian coordinates, but when transformed to the

space of Lie algebra (exponential coordinates), they follow a Gaussian distribution. Taking a cue from this research we posed the problem of pose and shape estimation of a medical snake robot in the space of the twist coordinates. We observed that the EKF used to find the pose and shape of the robot performed better than the one that used Euler angles and Cartesian coordinates for parameterization. However, using twist coordinates left the model highly non-linear and hence requiring linearization in the form of EKF.

An important connection we observed was that a Gaussian distribution in the space of $so(3)$ is a Bingham in the space of unit quaternion. We reformulated the problem of pose estimation by modeling the uncertainty in orientation using a Bingham distribution and translation using a Gaussian distribution. The resulting formulation performs filtering in a space where the probability distribution respects the constraints of the space. Further the Bingham distribution has two modes which captures the symmetric $\widetilde{\boldsymbol{q}} = -\widetilde{\boldsymbol{q}}$ property of unit quaternions. Our approach produces an objective function with a single basin of attraction when the correspondence is known. This allows us to guarantee global optimality. We validated our approach on a number of simulated and real world experiments. Our approach, however, is only locally optimal when the correspondence is unknown.

In order to deal with the nonconvexity of the problem introduced by lack of correspondence, we developed a multi-hypothesis method. The approach uses uncertainty in the pose to drive the exploration and exploitation, for finding the global optimum. This approach was inspired by particle filters, multi-hypothesis filters and branch-and-prune methods. While it is not guaranteed to find the global optimum, it proved to be very good at escaping from local minima, and requires tuning of very few parameters which can be tuned intuitively. We further extended our approach to deal with a small number of sparse point measurements. Our approach to sparse point registration can use as few as 20 measurements to accurately estimate the pose for a variety of shapes, as opposed to 100 measurements or more taken by most methods in literature.

Having developed a good understanding of probabilistic pose estimation as a MAP

problem, we decided to unify our understanding in a generalized framework. We developed a probabilistic framework that considers the problem of point-set registration in its most general form. By setting functional parameters on sensor noise, prior uncertainty, etc., our method captures most of the existing registration methods. This allows a user to intuitively choose the optimal approach for registration without having to try a large number of methods from literature. In addition, our work unifies disparate point-set registration approaches into a single probabilistic framework and reasons their application in terms of maximum likelihood estimate (MLE) or MAP estimate. Our framework also helps improve some of the existing registration methods, by theoretically highlighting some unstated assumptions and approximations made by existing methods.

Alongside the theoretical developments, there has been a constant emphasis in this thesis towards validating our findings on practical applications. Throughout the thesis one would find demonstration of our ideas on a number of simulation and real-world examples such as 3D reconstruction and object tracking in Chapter 6, hand-eye calibration in Chapter 4, touch-based localization in Chapter 8, tool-tip localization in Chapter 3, robot shape estimation in Chapter 5, stereo registration and image overlay in Chapter 10.

1.3 Technical Motivation

In this section, we present a list of key technical motivations for this thesis. While Sec. 1.2 describes the various needs resulting in the development of this thesis, this section consolidates the needs into technical motivations which then help highlight the key contributions in Sec. 1.4.

1.3.1 Obtaining a Linear Update Model for Pose Estimation

Typically, filtering-based approaches provide incremental pose estimates using non-linear update models that require linearization (in the case of EKF) or deterministic sampling (in the case of UKF), which make them highly sensitive to initial estima-

tion errors, and computationally expensive. This thesis will fulfill the need to develop linear update models which can ensure globally optimal estimates.

1.3.2 Using Appropriate Probability Distribution

The Gaussian distribution is a popular choice for modeling the uncertainty in pose parameters due to its convenient properties and natural appearance as a limit distribution [49, 79, 80, 64, 47]. However, naively choosing a Gaussian distribution may not capture the structure of error as it evolves and therefore an algorithm using such a distribution may produce low quality estimates [81]. Hence, this thesis will fulfill the need to develop probabilistic approaches for pose estimation using probability distributions that respect the structure of the parameter space.

1.3.3 Probabilistic Nonconvex Optimization for Pose Estimation

In applications such as point registration and 3D reconstruction, the correspondence between the measurements in the different reference frames are unknown resulting in a highly nonconvex problem [18]. Probabilistic methods that have been developed to deal with this nonconvexity use particle filters [16], simulated annealing [82], genetic algorithms [83] or multi-hypothesis filtering [84]. These implementations are computationally expensive and require several unintuitive parameters to be tuned. Thus, this thesis will attempt to develop filtering methods for nonconvex optimization that are computationally fast and have few intuitive parameters to tune.

1.3.4 Registration with Few Sparse Measurements

Several methods have been developed to perform registration when dense point measurements are obtained [11, 12, 22, 5, 23]. However, these methods do not perform well when the number of available point measurements are small (≈ 20), as in the case of probing-based registration in surgical applications [70]. Prior work assume *a priori* knowledge of landmarks or shape segments to hand-pick a small number of

probing locations [70, 71]; which can be a very restrictive assumption. Others use particle filter based approach which are computationally expensive and do not provide realtime pose updates [16]. Hence, this thesis will fulfill the need to develop a computationally fast approach to pose estimation in the presence of a small number of sparse measurements.

1.3.5 Unifying Registration Approaches in a Common Framework

Since registration is such a commonly encountered problem in a variety of fields, there are a number of different schools of thought when it comes to approaching this problem. While some consider it to be a sum of squares optimization [11, 27], others formulate it as a non-linear filtering problem [5, 6] or a mixed integer optimization problem [20]. Some groups iterate between correspondence matching and finding optimal pose [11, 18, 22, 23], while others optimize simultaneously for both correspondence and pose parameters [66, 20]. Even among the methods that explicitly reason about correspondence there is a rich diversity such as – closest point matching [11], point-to-plane matching [22], most likely matching [23, 21], and soft-matching [85]. This diversity in the methods makes it difficult for a researcher to pick the best approach that would work for their application. Often the easiest method to implement, such as ICP, may not yield optimal results for the chosen application. We believe that our generalized framework may guide the user into selecting which approach is best for a given problem.

1.4 Key Contributions

The key contributions of this thesis are summarized in Table 1.1.

Table 1.1: Motivation and Key Contributions of this Thesis

Technical motivation	Key contribution
Linear update model for pose estimation	Dual-quaternions for parameterizing pose
	The value of updating state using hashes of measurements
	Deriving uncertainty in the hash space
Using appropriate choice of probability distribution	One-to-one mapping between Gaussian in exponential coordinates and Bingham in unit quaternions
	A novel Bayes filter for pose estimation using pose, position, and surface-normal measurements
Probabilistic nonconvex optimization for registration	Multiple start branch and prune filter
Registration with few sparse measurements	Sparse point registration
Developing a unified framework for probabilistic registration	Unifying prior registration methods in literature
	Improving existing registration methods

1.4.1 Probabilistic Linear Models for Pose Estimation

In this work, we derive linear update models for pose estimation using dual quaternions to parameterize the pose. While dual quaternions have been used in the past for pose estimation, either linearizations were used [64] or measurement and pose uncertainties were ignored [60, 61]. In our work, we use pseudo-measurements which as obtained as a function of mini-batches of measurements (also referred to as 'hash' of measurements) to decouple rotation and translation and obtain a linear model. Further we use results from stochastic filtering theory to compute the uncertainty in the pseudo-measurements. To the best of our knowledge it is the first attempt at deriving linear update models for probabilistic pose estimation. In summary the key contributions resulting in probabilistic linear model are:

- Application of dual quaternions for probabilistic pose estimation.
- The value of updating using a hash of measurements.
- Deriving uncertainty in the hash space.

1.4.2 A Novel Bayesian Filter for Pose Update

The linear filter developed using dual quaternion, modeled the uncertainty in rotation quaternion using a Gaussian. The Gaussian distribution does not capture the constraints of the space of unit quaternions. Further Chirikjian *et. al.* [1] show that Gaussian distribution in the space of Lie algebra (Exponential coordinates) accurately capture the distribution of a stochastic system evolving on $SE(3)$. We observe than a Gaussian distribution in the space of exponential coordinates, $so(3)$, is a Bingham distribution in the space of unit quaternions. Thus, we improve upon our previous formulation by modeling the uncertainty in rotation quaternion using a Bingham distribution. The Bingham distribution is defined on a unit hypersphere hence capturing the unit norm constraint of the rotation quaternion. Further the Bingham distribution has two antipodal modes which captures the condition that unit quaternion $\widetilde{\boldsymbol{q}}$ and $-\widetilde{\boldsymbol{q}}$ represent the same rotation. Further we extend our formulation to process

not only position measurements, but also surface normal and pose measurements. In summary the key contributions of the Bingham filter for pose estimation are:

- One-to-one mapping between Gaussian distribution in exponential coordinates and Bingham distribution in unit quaternions.
- Updating pose using position, surface-normal and pose measurements

1.4.3 Multiple Start Branch and Prune Filter

We introduce a new probabilistic approach for nonconvex optimization, called multiple start branch and prune filter (MSBP). The MSBP starts with a number of initial states, similar to a multi-hypothesis filter. These states are perturbed based on the uncertainty. The perturbed states are updated and then pruned based on the innovation of the filter. This process is repeated iteratively until convergence. The perturbation step encourages exploration while the pruning step encourages exploitation. MSBP only has a few parameters to tune and can provide fast online estimates of the optimal pose. The key contributions of MSBP are

- A fully probabilistic approach to nonconvex optimization
- Exploration and escaping local minima is guided by state-uncertainty
- Exploitation and pruning of solutions stuck in local minima is performed using the innovation of the filter.

1.4.4 Probabilistic Sparse Point Registration

In this thesis, we develop a probabilistic method for robust sparse point registration (SPR) using a small batch of approximately 20 sensor measurements. Our approach 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 as initial guess to estimate the locally optimum registration. This process is repeated, until the pose converges within a tolerance bound. Similar

to MSBP, this approach is also probabilistic. However, a key difference from MSBP is that SPR processes a batch of small number of measurements, while MSBP processes a continuous stream of measurements. Key contributions of the SPR are:

- Registration estimate using as few as 20 sparse point measurements.
- Probabilistic batch-processing approach that does not use any heuristic information to form point correspondences.

1.4.5 Generalized Framework For Probabilistic Registration

An important contribution of this thesis is a probabilistic framework for point-set registration. Our framework poses registration as a maximum *A posteriori* estimate problem, taking into account uncertainties in the measurements as well as prior pose parameters. The two key contributions of this framework are:

- Unifying prior registration methods in literature under a common framework.
- Predicting improvements to existing registration methods using the framework.

1.5 Motivating Example in Medical Robotics

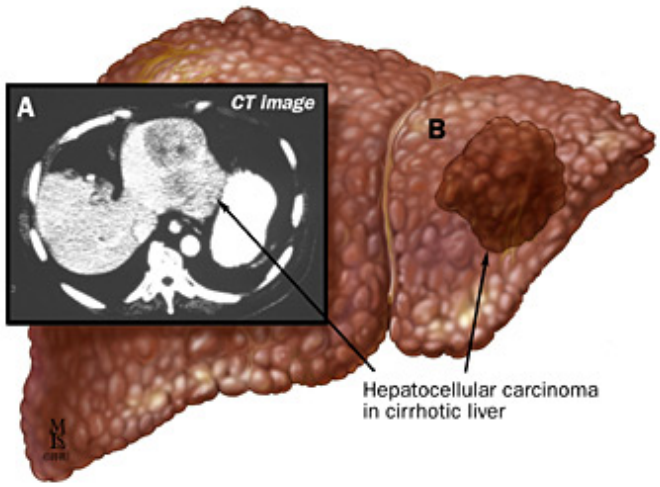


Figure 1-1: A. CT image of a liver with tumor. B. Corresponding preoperative model. Picture courtesy: Johns Hopkins Medicine Gastroenterology and Hepatology

The task at hand is to detect and localize a tumor in the liver of a patient using a robot assisted minimally invasive surgery. This task would first involve diagnosing the presence of tumor in the liver, which is typically done using imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI) or ultrasound (US). Fig. 1-1(A) shows a slice of the CT scan of the liver, in which the tumor is visible as a dark contrast. A preoperative model of the liver and the tumor is then generated from the CT scans (see fig. 1-1(B)).

A setup as illustrated in Fig. 1-2 is used to perform the surgery. The setup consists of a stereo vision system and a surgical robot with a force sensor attached to its tip. The following are some important problems to be solved in order to locate the tumor:

1. **Stereo registration**: Register a stereo reconstructed surface of the liver to its preoperative model; find pose $\boldsymbol{T}_C^M \in SE(3)$ in Fig. 1-2.

   The stereo camera provides a steady stream of thousands of point measurements, which need to be registered to the preoperative model. The pose estimation algorithm to be used needs to be capable of realtime computation, and be able to handle noise in the measurements. In addition, the algorithm would

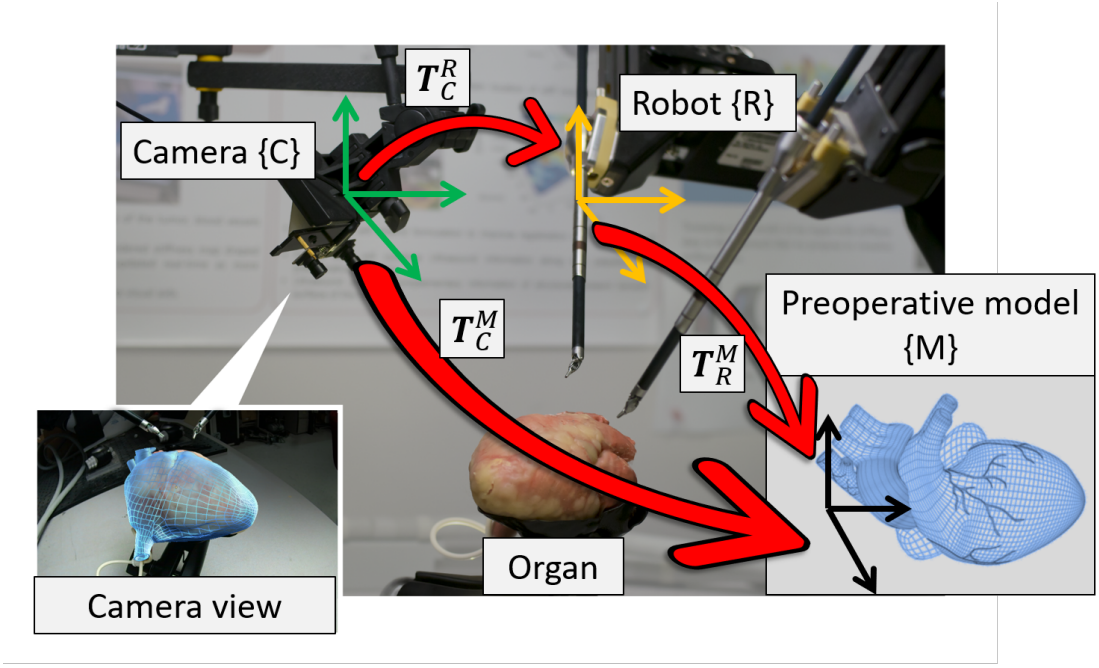


Figure 1-2: Experimental setup for the motivating example. There are three frames of reference that are of interest – robot frame R, camera frame C and preoperative model frame M. We need to find the pose among all these frames, $\boldsymbol{T}_R^M, \boldsymbol{T}_C^M, \boldsymbol{T}_C^R \in SE(3)$. This would allow us to virtually overlay the model of the tumor in the camera's view and help navigate the robot to the tumor location.

need to work without the knowledge of the point correspondence between the measurements and the preoperative model. Sequential pose estimation with unknown data association is described in Chapter 7. Once we perform stereo registration, the geometric model of the tumor can be augmented into the view of the camera. Augmented model of tumor would reduce the cognitive load of the surgeon and enable accurate extraction of the tumor. Chapter 10 describes the approach followed to perform the image overlay.

2. **Estimating pose between camera and robot**: We need to track the tip of the robot in the camera frame and in the robot frame, and find the relative pose between the two, $\boldsymbol{T}_C^R \in SE(3)$. Estimating the relative pose between the camera and the robot allows us to command the robot to a location as seen in the camera frame (see Fig. 1-2).

   The algorithm to be used for this problem needs to provide fast online updates of the pose along with an uncertainty measure to indicate convergence of the estimate. Unlike stereo registration, in this problem the data association can be trivially obtained by synchronizing the camera feed with the kinematics data. Chapter 6 and Chapter 4 provide details on how to estimate the pose for this case.

3. **Probing-based registration**: To improve upon the estimate of the stereo registration, the liver is probed with the robot and the obtained point measurements on the surface are registered to the preoperative model; we need to find $\boldsymbol{T}_R^M \in SE(3)$ in Fig. 1-2.

   In contrast to the stereo registration, there are fewer points available to be used in the pose estimation. The pose estimation algorithm to be used needs to be capable of using few sparse measurements to accurately register the robot frame to the model frame. Registration using sparse point measurements is described in Chapter 8.

4. **Deformation compensation**: Palpation introduces local deformation. This deformation if not compensated for can lead to erroneous pose estimation. The

deformation needs to be estimated and compensated for it during the registration.

The approach to be used would require realtime estimation of the local deformation from the sensed force and position measurements. Deformation compensated registration using complementary model update is described in Chapter 3.

Pose estimation is a common theme that binds all the problems listed above. However, each problem has unique constraints due to the nature of the measurements, the knowledge of the correspondences between the measurements, and the computation time requirements. In this thesis, we develop probabilistic means to estimate the pose for a variety of applications including the ones listed above. The approach that we follow provides fast and accurate estimates of the pose that is robust to noise in measurements and initial estimation errors.

Chapter 2

Mathematical background

2.1 Point-set Registration

Point-set registration is the process of finding a spatial transformation that aligns a point set defined in one frame of reference to a model (or another point set) defined with respect to a different reference frame. Registration is frequently encountered in robotic applications, such as computer vision [14], localization and mapping [15], surgical guidance [16], etc.

Horn's Method

When the correspondence between the points in the two point sets is known, rigid registration can be solved analytically using a least-squares formulation as shown in [4, 17, 60, 61]. Consider two point sets, $A = \{\boldsymbol{a}_i\}$, and $B = \{\boldsymbol{b}_i\}$, where $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$. Let $\boldsymbol{T} \in SE(3)$ be the transformation that aligns A and B,

$$\begin{pmatrix} \boldsymbol{a}_i \\ 1 \end{pmatrix} = \boldsymbol{T} \begin{pmatrix} \boldsymbol{b}_i \\ 1 \end{pmatrix}. \tag{2.1}$$

Note that the same transformation $\boldsymbol{T}$ aligns each $\boldsymbol{b}_i$ to $\boldsymbol{a}_i$. The objective function to be minimized is

$$O = \sum_{i=1}^{n} ||\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i - \boldsymbol{t}||^2, \tag{2.2}$$

where $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ are the rotation matrix and translation vector that comprise the transformation $\boldsymbol{T}$. Let the centroids of the two point sets be $\boldsymbol{a}_c$ and $\boldsymbol{b}_c$,

$$\boldsymbol{a}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{a}_i, \quad \boldsymbol{b}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{b}_i.$$

From Eq. 2.1,

$$\begin{aligned} \boldsymbol{a}_c &= \boldsymbol{R}\boldsymbol{b}_c + \boldsymbol{t}, \\ \Rightarrow \boldsymbol{t} &= \boldsymbol{a}_c - \boldsymbol{R}\boldsymbol{b}_c \end{aligned} \tag{2.3}$$

Eliminating $\boldsymbol{t}$ by substituting Eq. 2.3 in Eq. 2.2,

$$\begin{aligned} O &= \sum_{i=1}^{n} ||\boldsymbol{a}_i - \boldsymbol{R}\boldsymbol{b}_i + \boldsymbol{R}\boldsymbol{b}_c - \boldsymbol{a}_c||^2, \\ &= \sum_{i=1}^{n} ||(\boldsymbol{a}_i - \boldsymbol{a}_c) - \boldsymbol{R}(\boldsymbol{b}_i - \boldsymbol{b}_c)||^2, \\ &= \sum_{i=1}^{n} ||(\boldsymbol{u}_i) - \boldsymbol{R}(\boldsymbol{v}_i)||^2, \quad \text{where } \boldsymbol{u}_i = (\boldsymbol{a}_i - \boldsymbol{a}_c), \boldsymbol{v}_i = (\boldsymbol{b}_i - \boldsymbol{b}_c) \\ &= \sum_{i=1}^{n} \boldsymbol{u}_i^T \boldsymbol{u}_i + \boldsymbol{v}_i^T \boldsymbol{v}_i - 2(\boldsymbol{u}_i^T \boldsymbol{R}\boldsymbol{v}_i). \end{aligned} \tag{2.4}$$

We ignore the terms independent of $\boldsymbol{R}$. Minimizing O over $\boldsymbol{R}$ is equivalent to maximizing f,

$$\begin{aligned} f &= \sum_{i=1}^{n} (\boldsymbol{u}_i^T \boldsymbol{R} \boldsymbol{v}_i), \\ &= \mathrm{Tr}\left(\sum_{i=1}^{n} \boldsymbol{R} \boldsymbol{u}_i \boldsymbol{v}_i^T \right), \\ &= \mathrm{Tr}\left(\boldsymbol{R} \boldsymbol{N} \right), \quad \text{where } \boldsymbol{N} = \sum_{i=1}^{n} \boldsymbol{u}_i \boldsymbol{v}_i^T. \end{aligned} \tag{2.5}$$

The $\boldsymbol{R}$ that maximizes f can be obtained as

$$\begin{aligned} &\boldsymbol{R} = \boldsymbol{V} \boldsymbol{U}^T, \quad \text{where} \\ &\boldsymbol{N} = \boldsymbol{U} \boldsymbol{D} \boldsymbol{V}^T \quad \text{is the eigen decomposition of } \boldsymbol{N}. \end{aligned} \tag{2.6}$$

For detailed proof on why the optimal $\boldsymbol{R}$ would have the form as shown above, refer to [17].

Iterative Closest Point

However, when point correspondences are unknown, finding the optimal transformation becomes a nonconvex optimization problem with several local minima solutions. Besl *et. al.* came up with the popular iterative closest point (ICP) method that recursively finds correspondences and minimizes the alignment difference between point sets [11]. Over the years several variants of the ICP have been developed [12]. The ICP algorithm has two important steps:

1. Guess the correspondences between the point cloud and the model using a closest point rule.

2. Compute the transformation which minimizes the distance between corresponding points.

These two steps are repeated iteratively 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.

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

Algorithm 1: Iterative Closest Update

Consider two point clouds, $A = \{\boldsymbol{a}_i\}$, $\boldsymbol{a}_i \in \mathbb{R}^3$, $i = 1, \ldots, n$ are points on the geometric model of the object and $B = \{\boldsymbol{b}_j\}$, $\boldsymbol{b}_j \in \mathbb{R}^3$, $j = 1, \ldots, m$ are points obtained using sensor measurements. Let $\boldsymbol{T} \in SE(3)$ be the transformation that aligns A and B. The ICP algorithm is listed in Alg. 1. In Alg. 1, the minimization is typically performed using Horn's method [4]; although there are other optimization variants as well [17, 60, 61, 86].

2.2 Lie Group and Algebra

A Lie group is a set $\boldsymbol{G}$ with two structures: $\boldsymbol{G}$ is a group and $\boldsymbol{G}$ is a (smooth, real) manifold. These structures agree in the following sense: multiplication and inversion are smooth maps. This means that elements on the manifold also have a dual meaning of a group member, and since there exists an algebra that relates group elements to each other, then there is a corresponding relationship among the points on the manifold. A Lie algebra $\boldsymbol{g}$ associated to a Lie group $\boldsymbol{G}$ happens to be the tangent space at identity element $\boldsymbol{I}$.

Associated with the Lie group and its Lie algebra there are two important functions:

- Exponential map: This maps elements from the algebra to the manifold: exp : $\boldsymbol{g} \mapsto \boldsymbol{G}$.

- Logarithm map: This maps elements from the manifold to the algebra: log : $\boldsymbol{G} \mapsto \boldsymbol{g}$.

For a detailed discussion on Lie group and algebra, refer to [87, 88, 89, 90, 91, 92].

We provide the properties of the special Euclidean group $SE(3)$, which is an example of Lie group, that will be studied in this thesis. $SE(3)$ is the semi-definite product $\mathbb{R}^3 \times SO(3)$. The elements of $SE(3)$ can be represented by $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ in the form of the following 4×4 matrix,

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_{1\times 3} & 1 \end{pmatrix} \in SE(3). \tag{2.7}$$

For a vector $\boldsymbol{x} = [v_1, v_2, v_3, \omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^6$, an element $\hat{\boldsymbol{x}}$, of the Lie algebra $se(3)$ can be expressed as,

$$\hat{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{\Omega} & \boldsymbol{v} \\ \boldsymbol{0}_{1\times 3} & 0 \end{pmatrix} \in se(3), \tag{2.8}$$

where $\boldsymbol{v} = [v_1, v_2, v_3]^T$ and $\boldsymbol{\Omega}$ is the skew-symmetric matrix formed from $[\omega_1, \omega_2, \omega_3]^T$:

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{2.9}$$

The vector $\boldsymbol{x}$ is referred to as the twist vector (also referred to as exponential coordinates) and the operator $\hat{}$ is used to map from $\mathbb{R}^6$ to $se(3)$. The twist vector parametrizes an element of the Lie algebra $se(3)$ that belongs to the Lie group $SE(3)$. The element $\boldsymbol{T} \in SE(3)$, corresponding to $\hat{\boldsymbol{x}}$ can be obtained by using a matrix ex-

ponential,

$$\boldsymbol{T} = \exp(\hat{\boldsymbol{x}}), \tag{2.10}$$

and so the twist vector $\boldsymbol{x}$ is also referred to as exponential coordinates. Thus, given the exponential coordinates $\boldsymbol{x}$, the $SE(3)$ element can be obtained using the relation shown in Eq. 2.10.

2.3 Quaternion

While there are many representations for $SO(3)$ elements such as Euler angles, Rodrigues parameters, axis angles, etc, in this work uses unit-quaternions. We prefer the quaternions because their elements vary continuously over the unit sphere $\mathcal{S}^3$ as the orientation changes, avoiding discontinuous jumps (inherent to three-dimensional parameterizations) [93]. A quaternion $\widetilde{\boldsymbol{q}}$ is a 4-tuple:

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

where q_0 is the scalar part and $\text{vec}\,(\widetilde{\boldsymbol{q}}) = \boldsymbol{q} = (q_1, q_2, q_3)^T$ is the vector part of the quaternion. Sometimes an alternate convention is used where
$\widetilde{\boldsymbol{q}} = (q_1, q_2, q_3, q_0) = (vec(\widetilde{\boldsymbol{q}}), scalar(\widetilde{\boldsymbol{q}}))$ ([57]).

Quaternion Multiplication

Multiplication of two quaternions $\widetilde{\boldsymbol{p}}$ and $\widetilde{\boldsymbol{q}}$ is given by [59]

$$\widetilde{\boldsymbol{p}} \odot \widetilde{\boldsymbol{q}} = \underbrace{\begin{bmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & \boldsymbol{p}^\times + p_0 \boldsymbol{I}_3 \end{bmatrix}}_{\boldsymbol{F}_1(\widetilde{\boldsymbol{p}})} \boldsymbol{q} = \underbrace{\begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & -\boldsymbol{q}^\times + q_0 \boldsymbol{I}_3 \end{bmatrix}}_{\boldsymbol{F}_2(\widetilde{\boldsymbol{q}})} \boldsymbol{p} \tag{2.11}$$

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

Quaternion Conjugate

Given a quaternion $\widetilde{\boldsymbol{q}}$, its conjugate $\widetilde{\boldsymbol{q}}^*$ can be written as:

$$\widetilde{\boldsymbol{q}}^* = (q_0, -q_1, -q_2, -q_3)^T. \tag{2.12}$$

The norm of a quaternion is:

$$||\widetilde{\boldsymbol{q}}|| = \sqrt{\text{scalar}(\widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{q}}^*)}. \tag{2.13}$$

Unit Quaternion

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

$$\widetilde{\boldsymbol{q}} = \left(\cos\left(\frac{\theta}{2}\right), \boldsymbol{k}^T \sin\left(\frac{\theta}{2}\right)\right)^T. \tag{2.14}$$

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

Rotation Using Quaternions

A point $\boldsymbol{b}$ can be rotated by a quaternion $\widetilde{\boldsymbol{q}}$ to obtain a new point $\boldsymbol{a}$ as shown,

$$\widetilde{\boldsymbol{a}} = \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}} \odot \widetilde{\boldsymbol{q}}^*, \tag{2.15}$$

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

Properties of Quaternion Matrices

There are two important quaternion matrix functions, $\Omega_1 : \mathbb{R}^4 \to \mathbb{R}^{4\times4}$ and $\Omega_2 : \mathbb{R}^4 \to \mathbb{R}^{4\times4}$, which are defined as

$$\Omega_1(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_1^T & -q_2^T & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}, \quad \Omega_2(\boldsymbol{q}) = \begin{bmatrix} q_0 & -q_1^T & -q_2^T & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix},$$

for any unit quaternion $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T \in \mathbb{R}^4$. Some useful properties of these matrix functions are

1. $\Omega_2(\boldsymbol{q})\Omega_1(\boldsymbol{q})^T = \Omega_1(\boldsymbol{q})^T\Omega_2(\boldsymbol{q}) = \begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{R} \end{bmatrix}$, where $\boldsymbol{R} \in SO(3)$.
2. $\Omega_1(\boldsymbol{q})^T\Omega_1(\boldsymbol{q}) = \Omega_1(\boldsymbol{q})\Omega_1(\boldsymbol{q})^T = \Omega_2(\boldsymbol{q})^T\Omega_2(\boldsymbol{q}) = \Omega_2(\boldsymbol{q})\Omega_2(\boldsymbol{q})^T = (\boldsymbol{q}^T\boldsymbol{q})\boldsymbol{I}_{4\times4}$.
3. $\Omega_1(\boldsymbol{a})\boldsymbol{b} = \Omega_2(\boldsymbol{b})\boldsymbol{a}$.
4. $\Omega_1(\boldsymbol{a})^T\boldsymbol{a} = \Omega_2(\boldsymbol{a})^T\boldsymbol{a} = \boldsymbol{a}^T\boldsymbol{a}\boldsymbol{e}$, where $\boldsymbol{e} = (1, 0, 0, 0)^T$.

2.4 Dual Quaternion

There are many representations for $SE(3)$ elements such as Euler angles, quaternions, axis angles, etc. for rotation and Cartesian coordinates for translation. Dual quaternions compactly represent pose, and with the methods presented in this paper, give rise to a linear update model. Dual quaternions were introduced in 1873 by William Clifford [94] as a means to combine rotations and translations while retaining the benefits of the quaternion representation of rotation. Dual quaternions have traditionally been used for kinematic analysis of linkage mechanisms [95, 59, 58, 96]. Dual quaternions have recently gained popularity in graphics applications such as motion interpolation [97], transformation blending [98], as well as robotics applications such as hand-eye calibration [99, 32] and registration [64, 100, 101]. For a quick introductory tutorial on dual quaternions, refer to [102].

A dual quaternion $\hat{\boldsymbol{d}}$ is an 8-tuple $(p_0, p_1, p_2, p_3, q_0, q_1, q_2, q_3)$, which can be written in the form: $\hat{\boldsymbol{d}} = \widetilde{\boldsymbol{p}} + \epsilon\widetilde{\boldsymbol{q}}$, where $\widetilde{\boldsymbol{p}} = (p_0, p_1, p_2, p_3)$ and $\widetilde{\boldsymbol{q}} = (q_0, q_1, q_2, q_3)$ and quaternions and ϵ is a mathematical construct called the *dual operator* having the following property: $\epsilon \neq 0 \quad$ and $\epsilon^2 = 0$. The dual operator is a mathematical construct with a defined property and is not to be confused as having a small value close to 0. $\widetilde{\boldsymbol{p}}$ is called the real part and $\widetilde{\boldsymbol{q}}$ is called the dual part of the dual quaternion.

A dual quaternion used to represent a vector $\boldsymbol{a} \in \mathbb{R}^3$ has the following form

$$\hat{\boldsymbol{a}} = 1 + \epsilon\left(\widetilde{\boldsymbol{a}}\right), \quad \text{where } \widetilde{\boldsymbol{a}} = 0 + \boldsymbol{a}. \tag{2.16}$$

Dual Quaternion Multiplication

Multiplication of two dual quaternions $\hat{\boldsymbol{d}}_1 = \widetilde{\boldsymbol{p}}_1 + \epsilon\widetilde{\boldsymbol{q}}_1$ and $\hat{\boldsymbol{d}}_2 = \widetilde{\boldsymbol{p}}_2 + \epsilon\widetilde{\boldsymbol{q}}_2$ is given as

$$\hat{\boldsymbol{d}}_1 \otimes \hat{\boldsymbol{d}}_2 = \widetilde{\boldsymbol{p}}_1 \odot \widetilde{\boldsymbol{p}}_2 + \epsilon\left(\widetilde{\boldsymbol{p}}_1 \odot \widetilde{\boldsymbol{q}}_2 + \widetilde{\boldsymbol{q}}_1 \odot \widetilde{\boldsymbol{p}}_2\right), \tag{2.17}$$

where $\otimes$ is the dual quaternion multiplication operator.

Dual Quaternion Conjugate

Dual quaternions have three conjugates:

1. First conjugate: $\hat{\boldsymbol{d}}^{1*} = \widetilde{\boldsymbol{p}} - \epsilon\widetilde{\boldsymbol{q}}$. This definition is only provided for the sake of completeness, but is not used in this thesis.

2. Second conjugate: $\hat{\boldsymbol{d}}^{2*} = \widetilde{\boldsymbol{p}}^* + \epsilon\widetilde{\boldsymbol{q}}^*$. A dual quaternion is called "unit" if $\hat{\boldsymbol{d}} \otimes \hat{\boldsymbol{d}}^{2*} = 1$. This definition is important as the dual quaternion that parameterizes $SE(3)$ is a unit dual quaternion.

3. Third conjugate: $\hat{\boldsymbol{d}}^{3*} = \widetilde{\boldsymbol{p}}^* - \epsilon\widetilde{\boldsymbol{q}}^*$. An important property of the third conjugate that will be used in this work is, $\left(\hat{\boldsymbol{d}}_1 \otimes \hat{\boldsymbol{d}}_2\right)^{3*} = \hat{\boldsymbol{d}}_2^{3*} \otimes \hat{\boldsymbol{d}}_1^{3*}$.

Dual Quaternion for Pose Representation

A dual quaternion that is used to represent an $SE(3)$ element has the following form

$$\hat{\boldsymbol{d}} = \widetilde{\boldsymbol{q}}_r + \epsilon \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2}, \tag{2.18}$$

where $\widetilde{\boldsymbol{q}}_r$ is the rotation quaternion whose form is as shown in Eq. 2.14 and $\widetilde{\boldsymbol{q}}_t = 0 + \boldsymbol{t}$ is the quaternion representation of the translational component of the $SE(3)$ element, $\boldsymbol{t} \in \mathbb{R}^3$. For the sake of simplicity, we rewrite Eq. 2.18 as

$$\hat{\boldsymbol{d}} = \widetilde{\boldsymbol{q}}_r + \epsilon \widetilde{\boldsymbol{q}}_d, \quad \text{where} \tag{2.19}$$

$$\widetilde{\boldsymbol{q}}_d = \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2}. \tag{2.20}$$

It is important to note that $\hat{\boldsymbol{d}}$ is a unit dual quaternion since its dual-product with the second conjugate is unity. Let point $\boldsymbol{a} \in \mathbb{R}^3$ be obtained by transforming point $\boldsymbol{b} \in \mathbb{R}^3$ using a dual quaternion $\hat{\boldsymbol{d}}$. The transformation can be mathematically described as

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{d}} \otimes \hat{\boldsymbol{b}} \otimes \hat{\boldsymbol{d}}^{3*}, \tag{2.21}$$

where $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{b}}$ are obtained using Eq. 2.16.

Lemma 2.4.1. *For a unit dual quaternion, $\hat{\boldsymbol{d}} = \widetilde{\boldsymbol{q}}_r + \epsilon \widetilde{\boldsymbol{q}}_d$, the product of third and first conjugate equals unity: $\hat{\boldsymbol{d}}^{3*} \otimes \hat{\boldsymbol{d}}^{1*} = 1$.*

Proof:

$$\begin{aligned}
\hat{\boldsymbol{d}}^{3*} \otimes \hat{\boldsymbol{d}}^{1*} &= (\widetilde{\boldsymbol{q}}_r^* - \epsilon \widetilde{\boldsymbol{q}}_d^*) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon \widetilde{\boldsymbol{q}}_d) \\
&= \widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_r - \epsilon \left(\widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_d + \widetilde{\boldsymbol{q}}_d^* \odot \widetilde{\boldsymbol{q}}_r \right), \ \textit{from Eq. 2.11} \\
&= 1 - \epsilon \left(\widetilde{\boldsymbol{q}}_r^* \odot \frac{\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r}{2} + \frac{\widetilde{\boldsymbol{q}}_r^* \odot \widetilde{\boldsymbol{q}}_t^*}{2} \odot \widetilde{\boldsymbol{q}}_r \right).
\end{aligned} \tag{2.22}$$

Using the property that $\widetilde{\boldsymbol{q}}_r$ is a unit quaternion and $\widetilde{\boldsymbol{q}}_t^* = -\widetilde{\boldsymbol{q}}_t$ from Eq. 2.13. Eq. 2.22 can be further simplified as $\hat{\boldsymbol{d}}^{3*} \otimes \hat{\boldsymbol{d}}^{1*} = 1$.

2.5 Bingham Distribution

The Bingham distribution was introduced in [57] as an extension of the Gaussian distribution, conditioned to lie on the surface of a unit hyper-sphere. The Bingham

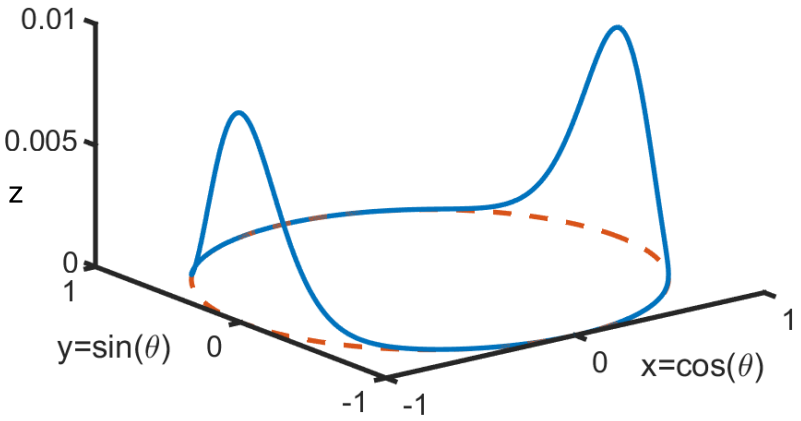


Figure 2-1: A 2D Bingham distribution: $z = \frac{1}{N}\exp(\boldsymbol{s}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{M}^T\boldsymbol{s})$, where $\boldsymbol{M} = \boldsymbol{I}_{2\times 2}$, $\boldsymbol{Z} = diag(0, -10)$, and $\boldsymbol{s} = (x, y)$. The mode is at $x = \pm 1, y = 0$.

distribution is widely used to analyze paleomagnetic data [103], computer vision [104] and directional statistics [57]. Recently the Bingham distribution has found applications in robotics for orientation estimation [81, 54], feature description [13] and planar pose description [55].

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

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

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

Mode of the Distribution

It can be shown that adding a multiple of the identity matrix $\boldsymbol{I}^{d\times d}$ to $\boldsymbol{Z}$ does not change the distribution ([57]).

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

where $N = \frac{N_1}{\exp(\lambda)}$, $\boldsymbol{x}^T\boldsymbol{x} = 1$ because $\boldsymbol{x}$ is on a unit hypersphere and $\boldsymbol{M}\boldsymbol{M}^T = \boldsymbol{I}$ by definition.

Thus, we conveniently force the first entry of $\boldsymbol{Z}$ to be zero ([57]). Because it is possible to swap columns of $\boldsymbol{M}$ and the corresponding diagonal entries in $\boldsymbol{Z}$ without changing the distribution, we can enforce $z_1 \geq \cdots \geq z_d$, where $z_1 = 0$. This representation allows us to obtain the mode of the distribution very easily by taking the first column of $\boldsymbol{M}$. The proof for this is shown below. The mode of $f(\boldsymbol{x})$ is given by,

$$\begin{aligned} \boldsymbol{x}^* &= \underset{\boldsymbol{x}}{\operatorname{argmax}}\, f(\boldsymbol{x}), \quad \text{subject to } ||\boldsymbol{x}|| = 1, \\ &= \underset{\boldsymbol{x}}{\operatorname{argmax}}\, \boldsymbol{x}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{M}^T\boldsymbol{x} - \lambda(\boldsymbol{x}^T\boldsymbol{x} - 1), \end{aligned} \tag{2.23}$$

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

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

Thus, the Lagrangian multiplier λ is the largest eigenvalue and $\boldsymbol{x}^*$ is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{M}\boldsymbol{Z}\boldsymbol{M}^T$. Note that we choose the largest eigenvalue because, when we substitute $\boldsymbol{M}\boldsymbol{Z}\boldsymbol{M}^T\boldsymbol{x} = \lambda\boldsymbol{x}$ in Eq. 2.23, $f(\boldsymbol{x}) = \lambda$.

And since we are maximizing $f(\boldsymbol{x})$ we choose the largest eigenvalue for λ. From the construction of $\boldsymbol{Z}$, the largest eigenvalue is 0 (which appears in the first column of $\boldsymbol{Z}$) and the corresponding eigenvector is the first column of $\boldsymbol{M}$.

Note that sometimes an alternate convention is used in literature, wherein $\boldsymbol{Z}$ is chosen such that the last entry of $\boldsymbol{Z}$ is 0 and the last column of $\boldsymbol{M}$ is chosen as the mode of the distribution ([81, 57]).

Normalization Constant

The normalization constant N is given by

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

The matrix $\boldsymbol{M}$ does not affect the normalization constant ([57]). An intuition for the same is, $\boldsymbol{Z}$ is an indicator of the spread of the distribution around the modes and $\boldsymbol{M}$ is an indicator of the location of the modes and the direction of the dispersion with respect to the modes. Hence, for a given $\boldsymbol{Z}$, changing $\boldsymbol{M}$ results in an identical distribution with a different location of the modes and orientation of the dispersion; which does not affect the integration. Fig. 2-2 shows two choices of $\boldsymbol{M}$ with the same $\boldsymbol{Z}$ that results in identical Bingham distributions that appear to be rotated with respect to each other.

Computation of the normalization constant is difficult and often one resorts to some form of approximation such as saddle point approximations, or precomputed lookup tables (see [13] and the references therein).

Antipodal Symmetry

An example of the PDF for two dimensions ($d = 3$) is shown in Fig. 2-1. The PDF's antipodal symmetry, i.e., $f(\boldsymbol{x}) = f(-\boldsymbol{x})$ holds for all $\boldsymbol{x} \in \mathcal{S}^{d-1}$. The antipodal symmetry is important when dealing with distribution of unit quaternions, because

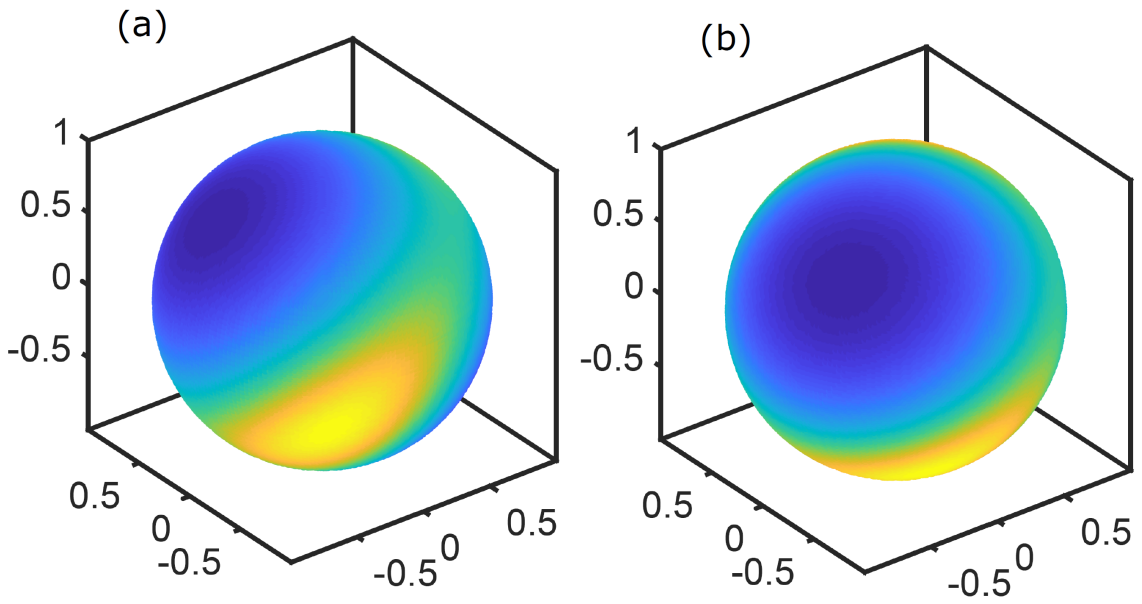


Figure 2-2: The figure shows two Bingham distributions with $\boldsymbol{Z} = diag(0, -0.5, -2)$, (a) $\boldsymbol{M} = \begin{pmatrix} 0.5322 & -0.4953 & -0.6866 \\ 0.7747 & 0.6120 & 0.1591 \\ 0.3415 & -0.6166 & 0.7094 \end{pmatrix}$, (b) $\boldsymbol{M} = \begin{pmatrix} 0.2603 & -0.6941 & -0.6712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

the $\widetilde{\boldsymbol{q}}$ and $-\widetilde{\boldsymbol{q}}$ describe the same rotation. The Bingham distribution with $d = 4$ is used to describe the uncertainty in the space of the unit-quaternions.

Product of two Bingham Densities

Similar to a Gaussian, the product of two Bingham PDFs is a Bingham distribution, which can be rescaled to form a PDF [81]. Consider two Bingham distributions $f_i(\boldsymbol{x}) = \frac{1}{N_i} \exp\left(\boldsymbol{x}^T \boldsymbol{M}_i \boldsymbol{Z}_i \boldsymbol{M}_i^T \boldsymbol{x}\right)$, $i = 1, 2$. Then,

$$\begin{aligned} & f_1(\boldsymbol{x}) \cdot f_2(\boldsymbol{x}) \\ & \quad = \frac{1}{N_1 N_2} \exp(\boldsymbol{x}^T \underbrace{\left(\boldsymbol{M}_1 \boldsymbol{Z}_1 \boldsymbol{M}_1^T + \boldsymbol{M}_2 \boldsymbol{Z}_2 \boldsymbol{M}_2^T\right)}_{\boldsymbol{A}} \boldsymbol{x}) \\ & \quad \propto \frac{1}{N} \exp\left(\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{Z} \boldsymbol{M}^T \boldsymbol{x}\right), \end{aligned} \tag{2.24}$$

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

Calculating the Covariance

Even though a Bingham distributed random vector $\boldsymbol{x}$ only takes values on the unit hyper-sphere, it is still possible to compute a covariance matrix in $\mathbb{R}^d$ ([81]), which is given by

$$Cov(\boldsymbol{x}) = E(\boldsymbol{x}^2) - E(\boldsymbol{x})^2, \tag{2.25}$$

$$= \boldsymbol{M}\left(diag\left(\frac{\frac{\partial N(\boldsymbol{Z})}{\partial \boldsymbol{z}_1}}{N(\boldsymbol{Z})}, \cdots, \frac{\frac{\partial N(\boldsymbol{Z})}{\partial \boldsymbol{z}_d}}{N(\boldsymbol{Z})}\right)\right)\boldsymbol{M}^T. \tag{2.26}$$

This is equivalently the covariance of a normally distributed $\boldsymbol{x}$ sampled from $\mathcal{N}\left(0, -0.5\left(\boldsymbol{M}(\boldsymbol{Z}+\lambda\boldsymbol{I})\boldsymbol{M}^T\right)^{-1}\right)$, given $|\boldsymbol{x}| = 1$, where $\lambda \in \mathbb{R}$ can be arbitrarily chosen as long as $(\boldsymbol{Z}+\lambda\boldsymbol{I})$ is negative definite ([54]). Fig. 2-3 shows two choices of λ resulting in the same Bingham distribution. Without loss of generality $\lambda = min(z_i)$ is chosen in this work.

Composition of two Bingham Distributions

Composition can be useful when we want to perturb a system, whose uncertainties are modeled with a Bingham distribution, with a Bingham distributed noise. Unfortunately, the Bingham distribution is not closed under composition and we can only approximate the composition as a Bingham ([57]). While the authors are not aware of a general approach for composing n dimensional Bingham distributions, for case of S^1 and S^3, prior work provides a method for finding the parameters of the approximate Bingham distribution obtained by composition [81] and [105].

2.6 Bayesian Filter

State estimation problems that utilize filtering algorithms are typically comprised of a recursive Bayesian formulation with probabilistic models. Bayesian filtering algorithms seek to estimate a posterior probability distribution over an unknown state vector $\boldsymbol{x}_k$ at time step k given the control inputs $\boldsymbol{u}_k$ and measurements $\boldsymbol{z}_k$. The PDF

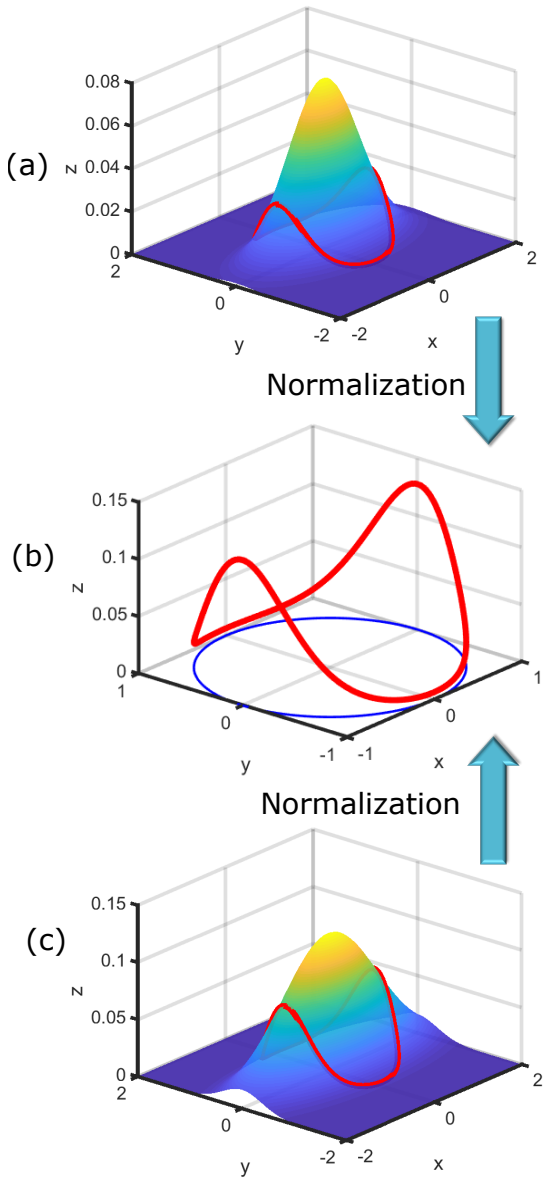


Figure 2-3: (a) A Gaussian distribution: $f_1(\boldsymbol{v}) = \frac{1}{N_1}\exp\left(\boldsymbol{v}^T\left(diag(0,-3) - 1\times\boldsymbol{I}_{2\times2}\right)\boldsymbol{v}\right)$. In red is shown a slice of the Gaussian as made by points lying on a unit circle. (b) A Bingham distribution: $f_2(\boldsymbol{v})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

can be factored using Bayes law,

$$p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = \eta p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k) p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k),$$

where η is a normalization constant. Assuming the measurements are independent and the processes Markov, we obtain

$$\begin{aligned} &p(\boldsymbol{x}_k|\boldsymbol{z}_{1:k}, \boldsymbol{u}_{1:k}) = \\ &\eta\ p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k) \int_{\boldsymbol{x}_{k-1}} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1}) p(\boldsymbol{x}_{k-1}|\boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1}) \partial \boldsymbol{x}_{k-1}. \end{aligned} \tag{2.27}$$

In Eq. 2.27, $p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k)$ is the measurement model and $p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1})$ is the process model. Eq, 2.27 is often written in the following form,

$$bel(\boldsymbol{x}_k) = \eta\ p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k)\ \overline{bel}(\boldsymbol{x}_k), \tag{2.28}$$

where $bel(\boldsymbol{x}_k)$ is the belief over the state $\boldsymbol{x}_k$ and $\overline{bel}(\boldsymbol{x}_k)$ is the posterior before incorporating measurement $\boldsymbol{z}_k$.

Kalman Filter

The Kalman filter is a Bayesian filter, which assumes that $\boldsymbol{x}_k$, $\boldsymbol{z}_k$ and $\boldsymbol{u}_k$ are Gaussian distributions. If the system has a linear measurement model and a linear process model, the Kalman filter is the optimal stochastic estimator for the state $\boldsymbol{x}_k$ [51]. The posterior distribution $\boldsymbol{x}_k$ is parameterized as

$$\boldsymbol{x}_k \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k).$$

There are two steps in a Kalman filter: prediction and update. In the prediction step, given the previous estimate of the state, the current state is estimated using the

process model. From Eq. 2.28,

$$\begin{aligned}
\overline{bel}(\boldsymbol{x}_k) &= \int p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_{1:k-1}) bel(\boldsymbol{x}_{k-1}) \partial \boldsymbol{x}_{k-1}, \\
&= \eta_1 \int \exp\left(-\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{A}_k\boldsymbol{x}_{k-1} - \boldsymbol{B}_k\boldsymbol{u}_k)^T \boldsymbol{R}_k^{-1}(\boldsymbol{x}_k - \boldsymbol{A}_k\boldsymbol{x}_{k-1} - \boldsymbol{B}_k\boldsymbol{u}_k)\right) \\
&\exp\left(-\frac{1}{2}(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1})^T \boldsymbol{\Sigma}_{k-1}^{-1}(\boldsymbol{x}_{k-1} - \boldsymbol{\mu}_{k-1})\right) d\boldsymbol{x}_{k-1}, \\
&= \eta_1 \exp\left(-\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})^T \boldsymbol{\Sigma}_{k|k-1}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})\right), \quad \text{where} \\
\boldsymbol{\mu}_{k|k-1} &= \boldsymbol{A}_k\boldsymbol{\mu}_{k-1} + \boldsymbol{B}_k\boldsymbol{u}_k, \qquad (2.29) \\
\boldsymbol{\Sigma}_{k|k-1} &= \boldsymbol{A}_k\boldsymbol{\Sigma}_{k-1}\boldsymbol{A}_k^T + \boldsymbol{Q}_k, \qquad (2.30)
\end{aligned}$$

where $\boldsymbol{A}_k$ is the state transition matrix, $\boldsymbol{B}_k$ is the control input matrix and $\boldsymbol{Q}_k$ is the covariance of the process model noise.

The update step corrects the predicted estimate using the obtained sensor measurement $\boldsymbol{z}_k$ by computing a Kalman gain $\boldsymbol{K}_k$. To obtain the update equations we use Eq. 2.28

$$\begin{aligned}
bel(\boldsymbol{x}_k) &= \eta\ p(\boldsymbol{z}_k|\boldsymbol{x}_k, \boldsymbol{z}_{1:k-1}, \boldsymbol{u}_k)\overline{bel}(\boldsymbol{x}_k), \\
&= \eta \exp\left(-\frac{1}{2}(\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{x}_k)^T \boldsymbol{R}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{x}_k)\right) \\
&\exp\left(-\frac{1}{2}(\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})^T \boldsymbol{\Sigma}_{k|k-1}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})\right),
\end{aligned}$$

where $\boldsymbol{H}_k$ is the measurement model, such that $\boldsymbol{z}_k = \mathcal{N}(\boldsymbol{H}_k\boldsymbol{x}_k, \boldsymbol{R}_k)$ and $\boldsymbol{R}_k$ is the covariance of the measurement noise. We can obtain the $\boldsymbol{\mu}_{k|k}$ by finding the maximum likelihood estimate of $bel(\boldsymbol{x}_k)$

$$\begin{aligned}
\boldsymbol{\mu}_{k|k} &= \underset{\boldsymbol{x}_k}{\operatorname{argmax}}\, bel(\boldsymbol{x}_k), \\
&= \underset{\boldsymbol{x}_k}{\operatorname{argmin}} \left((\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{x}_k)^T \boldsymbol{R}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{x}_k) + (\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})^T \boldsymbol{\Sigma}_{k|k-1}^{-1}(\boldsymbol{x}_k - \boldsymbol{\mu}_{k|k-1})\right), \qquad (2.31)
\end{aligned}$$

where $\boldsymbol{H}_k$ is the measurement model, such that $\boldsymbol{z}_k = \mathcal{N}(\boldsymbol{H}_k\boldsymbol{x}_k, \boldsymbol{R}_k)$ and $\boldsymbol{R}_k$ is the

covariance of the measurement noise. Upon simplification we obtain

$$\boldsymbol{\mu}_{k|k} = \boldsymbol{\mu}_{k|k-1} + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\boldsymbol{\mu}_{k|k-1}), \tag{2.32}$$

$$\boldsymbol{\Sigma}_{k|k} = \boldsymbol{\Sigma}_{k|k-1} - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{\Sigma}_{k|k-1}, \quad \text{where,} \tag{2.33}$$

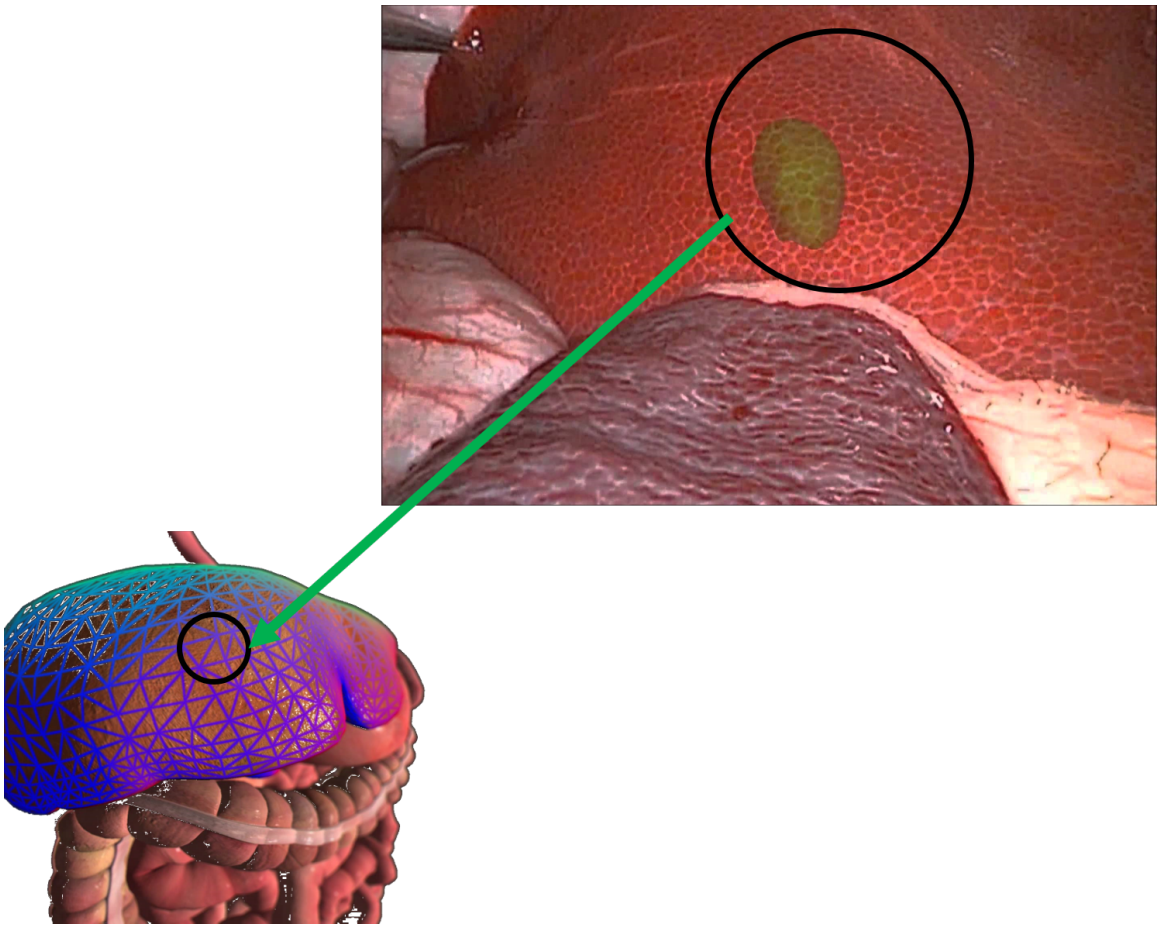
$$\boldsymbol{K}_k = \boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{\Sigma}_{k|k-1}\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}. \tag{2.34}$$

From the above equations it can be observed that the uncertainty of the state $\boldsymbol{\Sigma}_{k|k}$, is reduced upon the inclusion of the information provided by the measurement.

The Kalman filter is only optimal for linear systems. For systems with nonlinear process models and nonlinear measurement models, one popular method is to use an EKF. An EKF linearizes the models about the current estimate and then perform a similar prediction and measurement correction step. The other popular method is to use an UKF. The UKF uses a deterministic sampling technique known as the unscented transform to pick a minimal set of sample points (called '*sigma points*') around the mean. These sigma points are then propagated through the nonlinear functions, from which a new mean and covariance estimate are then formed. The result is a filter which, for certain systems, more accurately estimates the true mean and covariance. In addition, this technique removes the requirement to explicitly calculate Jacobian (as required by EKF), which for complex functions can be a difficult task in itself, if not impossible.

Part I

Using mini batches of measurements for state update



Chapter 3

Complementary Model Update

Minimally invasive surgery (MIS) has the potential to reduce patient trauma, post-operative complications, recovery time and cost. Unfortunately, surgical navigation remains a challenging task in MIS, and hence computer aided surgery (CAS) was developed to assist surgeons in overcoming this challenge. A prerequisite for successful CAS is an accurate registration of preoperative images to intra-operative anatomy. Typically, prior work relied on image guided techniques [106] while others used structured light [107] and force-based exploration [108, 109]. In this chapter, we pursue the concept of sensory-guided surgery based on force/contact exploration in flexible environments.

Prior work by Sanan *et. al.* [78] addressed simultaneous compliance and registration estimation (SCAR) using a filtering approach. However, the filtering implementation for SCAR is not robust to initialization error, and therefore we have developed a new update model, termed complementary model update (CMU) that can be used to accurately estimate the registration, in a filtering approach as well as using a generic non-linear optimizer. This update model encodes both contact force and contact location information.

The use of contact/force data comes with the advantage of providing stiffness information, which can enhance the performance and robustness of registration. This advantage, however, comes at a cost of having to consider the induced local surface deformation due to the physical interaction of the surgical tool with the organ. In

this work, we simultaneously use multiple position-force measurements to estimate the local deformation, which is then used to estimate the local stiffness as well as registration parameters. Prior work in literature has looked at different types of palpation strategies to infer mechanical properties of the tissue. For example, Yamamoto *et. al.* [109] perform discrete probing of the tissue, while Liu *et. al.* [110] perform a continuous motion palpation and Goldman *et. al.* [108] perform dynamic excitation of the tissue. While the above mentioned approaches perform different types of mechanical palpation to find tissue stiffness, they restrict themselves to flat organs or assume they are already registered to the preoperative model. An exception being the work of Chalasani *et. al.* [111, 112], who assume non-flat organs and estimate the shape of the organ and stiffness simultaneously using the force-position measurements, but do not register themselves to a preoperative model. Since the publication of our work [113], there has been more recent work by Wang *et. al.* [114] which uses contact measurements to perform deformable registration, but does not estimate stiffness in the process.

The performance of the our formulation is evaluated for various geometries, stiffness distributions, subject to sensor noise levels and initial conditions. The new formulation is validated with simulated data and experimental data obtained by palpating a silicone phantom organ and an *ex vivo* organ. A simple experimental setup consisting of a surgical tool that can sense force and position at its tip is used to validate the core ideas for the use of contact/force based exploration data for registration. The robot experiments in this chapter were carried out by our collaborators at Vanderbilt university. Ultimately, we show that the CMU approach overcomes the aforementioned issues associated with local deformations due to contact exploration and offers improved performance compared to other known methods. We believe this approach can be used in MIS for resection or ablation of tumors in organs such as liver, bladder and kidneys [115].

3.1 Related Work

When the anatomy is rigid, registration yields the homogeneous transformation matrix that relates points in the anatomy's frame to the frame of its *a priori* model. When dealing with anatomy that is flexible, we observe discrepancy between the model of the source and the target due to– (i) the global deformation of the target caused by swelling or organ shift due to gravity and positioning of the patient, (ii) local deformation due to tool-tissue interaction. Non-rigid registration techniques, that are popularly referred to as "deformable registration" methods, have been developed to address this issue. A popular method used to perform deformable registration is the coherent point drift (CPD) developed by Myronenko *et. al.* [116]. CPD uses a probabilistic matching approach and assumes motion coherence between the points in the two frames to find the registration as well as the optimal deformation of the model. The elastic convolved ICP by Sagawa *et. al.* [117] uses organ geometry and image processing to perform registration, but performs poorly when the visible organ is obfuscated with blood and also respond adversely to change in lighting. Sotiras *et. al.* [118] in their review paper describe various other deformable registration methods that are commonly used in literature. The majority of previous works relies on non-contact based methods to produce geometric data for registration.

Other imaging modalities such as intraoperative ultrasound (US) based registration for soft bodies deal with registering the preoperative model to the US image, instead of the surgical tool [119, 120]. The approach of Lunn *et. al.* [121] requires a pre-registration step where either fiducial markers are used or an expert manually chooses points of interest from the US image to provide a good initial guess for the registration algorithm. The approach of Songbai *et. al.* [122] on the other hand, present a fiducial free way for registration using 3D ultrasound (3DUS). But this method depends heavily on finding distinctive 3DUS features which may not be successful in every surgical scenario. To the best of our knowledge, the only work that uses contact/force based blind exploration data to perform registration is the work of Sanan *et. al.* [78], who used an IEKF to simultaneously estimate the registration

parameters and generate a stiffness map of the environment. In Section 3.2 we discuss some of the drawbacks of the IEKF implementation of SCAR and motivate the need for a more robust formulation.

Problem statement and assumptions

Given an *a priori* geometric model of an organ as well as the measurements of the tool tip positions and associated contact forces, (i) the surgical tool needs needs to be registered to the frame of the model, and (ii) the stiffness distribution over the organ's surface needs to be estimated. We make the following assumption in this chapter:

1. The true shape of the organ is not globally deformed but instead experiences local deformations only due to interaction with the tool.

2. The tool-tip's position can be measured accurately.

3. The tool has force sensing capability so that it can be servoed in a hybrid position-force control manner.

4. The forces applied by the tool are within the admissible range (typically $\approx 1N$) in which the organ only undergoes a small deformation that allows it to realize its undeformed state when the force is removed.

5. The friction between the tool tip and the surface of the soft body is negligible.

3.2 Problem Formulation

In this section, we describe the complementary model update (CMU) to simultaneously estimate the variation of stiffness over the surface as well as register a flexible environment to its *a priori* model. Our group had earlier developed a filtering approach for SCAR that was reported in [78]. For the rest of this chapter, the old implementation of SCAR using IEKF will be referred to as SCAR-IEKF-old. SCAR-IEKF-old

Table 3.1: Notations

Symbol	Description
$[\cdot]^R$	Entities defined in robot's frame
$[\cdot]^C$	Entity defined in model's frame
$\boldsymbol{n}$	Normal vector
${}^d\boldsymbol{p}$	Coordinates of deformed point
${}^u\boldsymbol{p}$	Coordinates of undeformed point
$\boldsymbol{T}$	Homogeneous transformation matrix
f	Force magnitude
c	Stiffness
d	Deformation depth
$\boldsymbol{\phi}$	CAD model

uses a geometric prior represented in the form of a triangular mesh and therefore each triangle was assigned its own stiffness values. The state vector $\boldsymbol{x}_k$ consisted of six registration parameters and the stiffness values associated with each triangle of the triangular mesh. The update step involved using sensed position and force measurements to minimize the following objective function:

$$h_1(\boldsymbol{x}_k) = -(\boldsymbol{n}_j^C)^T \left({}^d\boldsymbol{p}_j^C - {}^u\boldsymbol{p}_j^C\right)(c_i)_{\boldsymbol{x}_k} + f_j, \tag{3.1}$$

$$\text{where } {}^d\boldsymbol{p}_j^C = \boldsymbol{T}_{\boldsymbol{x}_k}\left({}^d\boldsymbol{p}_j^R\right). \tag{3.2}$$

The position of the j^{th} deformed point is ${}^d\boldsymbol{p}_j^R$, measured in the robot frame[1] and the corresponding sensed force is f_j. The corresponding closest point on the CAD model is ${}^u\boldsymbol{p}_j^C$ and the normal vector is $\boldsymbol{n}_j^C$. The objective function $h_1(\boldsymbol{x}_k)$ is the difference between the estimated force and the measured force. The objective function is minimized over $\boldsymbol{x}_k$ and as mentioned above, $\boldsymbol{T}$, the homogeneous transformation matrix and c_i, the stiffness associated with triangle i (see notations in Table 3.1) are obtained from $\boldsymbol{x}_k$ and hence are updated simultaneously.

Fig. 3-1 shows the true location of the deformed point ${}^{d^*}\boldsymbol{p}_1^C$ which is at a depth d^* along the normal $\boldsymbol{n}_1^C$. Let the location of the deformed point obtained from the registration estimate of previous iteration be ${}_{k-1}{}^d\boldsymbol{p}_1^C$ as shown in Fig. 3-1. The subscript

[1]We assume that the tool is rigidly attached to the robot and hence transformation between the robot's frame and the tool's frame can be carried out trivially by a precomputed rigid transformation.

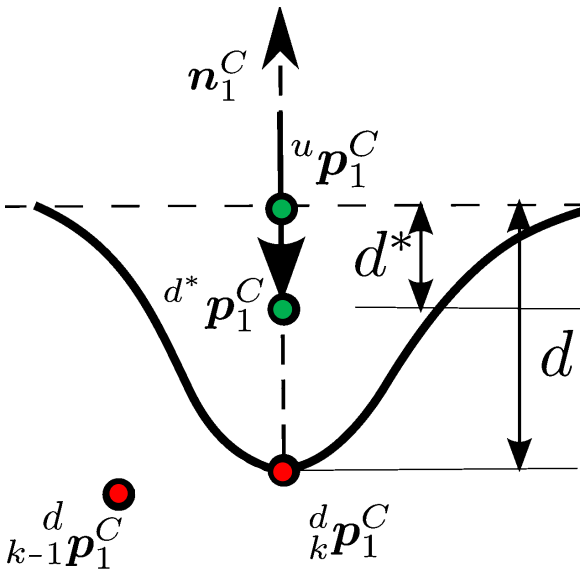


Figure 3-1: Schematic shows ambiguity in single measurement based update

$k-1$ indicates that the coordinates are transformed using the registration estimate from the previous iteration. After applying the state update using Eq. 3.1, the updated position of the deformed point is $^{d}_{k}\boldsymbol{p}_1^C$, which is at a depth d along the normal (see Fig. 3-1). The filter estimates the stiffness value to be $c_i = \frac{f_j}{d} \neq \frac{f_j}{d^*} = c_i^*$. As can be observed from Eq. 3.1, substituting c_i and $^{d}_{k}\boldsymbol{p}_1^C$ yields $h_1(\boldsymbol{x}_k) = 0$. Substituting c_i^* and $^{d^*}_{k}\boldsymbol{p}_1^C$ in Eq. 3.1 also yields $h_1(\boldsymbol{x}_k) = 0$. This results in an ambiguity in registration along the normal and an incorrect stiffness estimation. We make an observation that when the registration is updated based on at least a pair of observations, the ambiguity in registration is resolved. Let us select two points, $^{u}\boldsymbol{p}_1^C$ and $^{u}\boldsymbol{p}_2^C$, on the undeformed surface of the organ that are spatially close to each other. Since the points are close to each other, we assume that the normals n^C and the stiffness c at both locations are the same. Let us apply a force of magnitude f_1 and f_2 respectively at $^{u}\boldsymbol{p}_1^C$ and $^{u}\boldsymbol{p}_2^C$. Upon application of the force, the surface would deform by depths:

$$d_i = \frac{f_i}{c}, \quad i = 1, 2. \tag{3.3}$$

From Eq. 3.3, we have:

$$c = \frac{f_2 - f_1}{d_2 - d_1}, \quad \text{when } d_2 \neq d_1. \tag{3.4}$$

Let the coordinates of the deformed points be ${}^{d}p_1^R$ and ${}^{d}p_2^R$. From Eq, 3.4, stiffness $c = \frac{f_2 - f_1}{||{}^{d}\boldsymbol{p}_2^R - {}^{d}\boldsymbol{p}_1^R||}$. We can now relate the deformed and undeformed probed points from the linear stiffness model (see Fig. 3-1):

$$\begin{aligned} {}^{u}\boldsymbol{p}_1^C - \boldsymbol{n}^C d_1 &= {}^{d}\,\boldsymbol{p}_1^C \\ \Rightarrow {}^{u}\boldsymbol{p}_1^C - \boldsymbol{n}^C \frac{f_1}{c} &= \boldsymbol{T}({}^{d}\boldsymbol{p}_1^R), \end{aligned} \tag{3.5}$$

where $\boldsymbol{T}(\boldsymbol{p})$ transforms $\boldsymbol{p}$ from tool-frame to CAD model-frame. The LHS of Eq. 3.5 is the estimated position of the deformed point in the CAD model frame based on the estimated stiffness c and the RHS is the coordinates of the sensed deformed points transformed to the CAD model frame. While we used two points to estimate the stiffness c in Eq. 3.4, in general we can use l measurements, where $l \geq 2$, to estimate the stiffness:

$$\begin{aligned} c &= \frac{f_k}{d_k}, k = 1, 2, ...l \\ \Rightarrow c &= \frac{f_i - f_j}{d_i - d_j}, \quad i, j \in \{1, 2, ..., l\} \text{ and } i \neq j. \end{aligned} \tag{3.6}$$

From Eq. 3.6, stiffness c linearly relates the depth $(d_i - d_j)$ to the applied force $(f_i - f_j)$. Thus we have $c = L(d_i - d_j, f_i - f_j)$, where L is the function that returns the slope of the best line fit through a regression on the data $\{(d_i - d_j), (f_i - f_j)\}$.

Based on Eq. 3.5, we can form a new objective function for obtaining the best registration as follows:

$$h_2(\boldsymbol{T}) = \sum_{j=1}^{m} \left\| {}^{u}\boldsymbol{p}_j^C - \frac{\boldsymbol{n}_j^C (f_\beta)_j}{c_j} - \boldsymbol{T}(({}^{d}\boldsymbol{p}_\beta^R)_j) \right\|^2, \tag{3.7}$$

where $(f_\beta)_j$ and $({}^{d}\boldsymbol{p}_\beta^R)_j$ are the force and position measurements obtained by palpating

the j^{th} undeformed point and m is the total number of undeformed points being probed. $\beta \in \{1, 2, ..., l\}$ $(l \geq 2)$ is the index of the measurement taken from the set of measurements at the j^{th} undeformed point.

The stiffness is estimated first from the linear regression L using force-position measurements and then used to optimize $h_2(\boldsymbol{T})$. The objective function $h_2(\boldsymbol{T})$ is the difference in the squared norm of the distance between the estimated location of the deformed point and the sensed location of the deformed point.

It is to be noted from Eq. 3.7 that $h_2(\boldsymbol{T})$ only updates the registration and Eq. 3.4 updates the stiffness, unlike $h_1(\boldsymbol{x}_k)$ that is used to update both stiffness and registration (see Eq. 3.1). The decoupled stiffness and registration update implies that error in registration does not affect stiffness estimate unlike in the work of Sanan *et. al.* [78]. Fig. 3-2 shows a flowchart that provides an overview of the various steps described so far.

The various steps involved in the CMU approach are listed below:

1. *Collection*: In the collection step, pairs of force-position measurements which satisfy the following conditions are grouped together in the same set:

   (a) The force magnitudes are different.

   (b) The direction of normals fall within a threshold of each other.

   (c) The position measurements fall within a threshold of each other.

   The three conditions stated above imply that position measurements correspond to the same undeformed point being probed with different forces, and forming a compatible set. Given the measurements $({}^d\boldsymbol{p}_i^R, f_i)$, $i = 1, 2, ..., n$ obtained so far, we collect compatible sets, $\{(({}^d\boldsymbol{p}_1^R)_j, (f_1)_j), (({}^d\boldsymbol{p}_2^R)_j, (f_2)_j), ...\}$, $j = 1, 2, ..., m$, where m is the total number of distinct sets obtained.

2. *Stiffness estimation*: For each of the compatible sets that have at least one pair of force-position measurements, we estimate the local stiffness assuming a linear stiffness model as shown in Eq. 3.6. In Sec. 3.5, we refer to the process of using a criteria to collect a subset of measurements and then using those

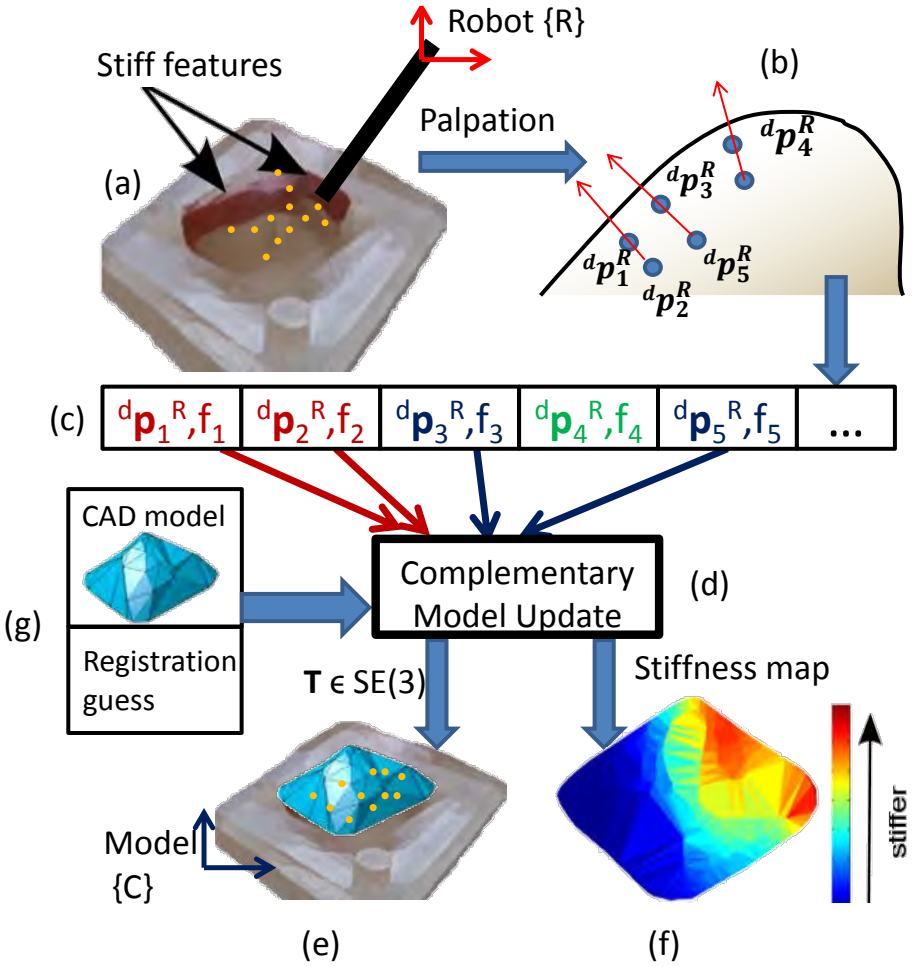


Figure 3-2: Flowchart describing the inputs and outputs for complementary model update (a) Flexible environment with embedded stiff features is probed by a robot (b) Location of probed points are sensed (c) Compatible force-position measurements are collected (d) complementary model update estimates the registration and stiffness map (e) Robot frame and model frame are registered (f) Stiffness map is generated (g) Prior geometric model and the initial registration guess

measurements to estimate the stiffness as *hashing* of measurements. The linear regression used to find the stiffness is an example of a *hash function*[2]. And the estimated stiffness is referred to as a *pseudo measurement*.

3. *Correspondence*: The points $({}^{d}\boldsymbol{p}^{R}_{\beta})_j$ are transformed using the best registration estimate available to obtain $({}^{d}\boldsymbol{p}^{C}_{\beta})_j$. We then find, $\left({}^{u}\boldsymbol{p}^{C}_{j}, \boldsymbol{n}^{C}_{j}\right) = M\left(({}^{d}\boldsymbol{p}^{C}_{\beta})_j, \boldsymbol{\phi}\right)$, where M is the rule that finds the closest point ${}^{u}\boldsymbol{p}^{C}_{j} \in \boldsymbol{\phi}$ to $({}^{d}\boldsymbol{p}_{\beta})^{C}_{j}$ and the corresponding normal n^{C}_{j}. Other alternates for M include methods such probabilistic most likely correspondence as developed by Billing *et. al.* [23], probabilistic multi-point correspondence as in the case of coherent point drift of Myronenko *et. al.* [116] or maximum penalized likelihood estimation of Kang *et. al.* [124], and similarity transformation coefficients of Ryan *et. al.* [125].

4. *Minimization*: The objective function described in Eq. 3.7 is minimized using a least squares solver, such as Horn's method [4] or can be used in the update step of a Kalman filter (for example a UKF for registration by Moghari *et. al.* [5]) to estimate the registration.

5. We loop between the *Correspondence* and *Minimization* step until convergence or upto a fixed number of iterations, upon obtaining $\boldsymbol{T}$.

We present results from an implementation of CMU that uses a least squares optimizer for minimization [17]. We retain the convention used earlier to refer to our implementation as SCAR-LSQ-CMU. We have also implemented a filtering approach using CMU, which will be referred to as SCAR-IEKF-CMU.

The minimization step would only return a local minima when using a filter or least squares solver. One way to find global minima is to use a branch and bound technique as described by Yang *et. al.* [18].

[2]It must be noted that this is not the standard usage of the term hash function. The formal definition of a hash function is any function that can be used to map data of arbitrary size to data of fixed size [123]. While hash functions are typically used for accelerated database lookup, we borrow the terminology in this work to denote the function that takes in measurements of arbitrary size and returns a pseudo-measurement of fixed size.

Input:
${}^d\boldsymbol{p}_i^R \in \boldsymbol{R}^3, i = 1, 2, ..., n$
$f_i \in \boldsymbol{R}, i = 1, 2, ..., n$
A priori CAD model: $\boldsymbol{\phi}$
Initial transformation: $\boldsymbol{T}_0 \in \boldsymbol{SE}(3)$
Output: Transformation $\boldsymbol{T}$ that aligns ${}^d\boldsymbol{p}_i^R$ with $\boldsymbol{\phi}$
Collection: Collect points satisfying compatibility criteria:
$\left\{\left(({}^d\boldsymbol{p}_1^R)_j, (f_1)_j\right), \left(({}^d\boldsymbol{p}_2^R)_j, (f_2)_j\right), ...\right\}$
Stiffness estimation: Estimate the linear stiffness
$c_j = L\left(\left\|({}^d\boldsymbol{p}_\beta^R)_j - ({}^d\boldsymbol{p}_\gamma^R)_j\right\|, ((f_\beta)_j - (f_\gamma)_j)\right)$ **Optimize** *Initialize*: $\boldsymbol{T} \leftarrow \boldsymbol{T}_0$
while *not converged* **do**
  Correspondence: $\left({}^u\boldsymbol{p}_j^C, \boldsymbol{n}_j^C\right) = M\left(\boldsymbol{T}({}^d\boldsymbol{p}_\beta^R)_j, \boldsymbol{\phi}\right)$
  Minimization: $\boldsymbol{T} = \underset{\boldsymbol{T}}{\operatorname{argmin}} \sum_{j=1}^{m} \left\| {}^u\boldsymbol{p}_j^C - \frac{\boldsymbol{n}_j^C (f_\beta)_j}{c_j} - \boldsymbol{T}({}^d\boldsymbol{p}_\beta^R)_j \right\|^2$
end

Algorithm 2: Complementary Model Update

3.3 Results: Batch Estimation with Unknown Data Association

3.3.1 Comparison with old implementation of SCAR

In Fig. 3-3 we compare the results of SCAR-LSQ-CMU with SCAR-IEKF-old for a simulated case of a pyramid shaped organ with an embedded stiff feature that was probed at 250 uniformly spaced points. The CAD model of the organ is represented in the form of a triangular mesh with 524 faces. We choose an initial guess for registration which is displaced along the Z direction by 4mm.

SCAR-LSQ-CMU accurately estimates the true registration after palpation of about 20 points. While SCAR-IEKF-old correctly estimates the translation components of registration (albeit not better than SCAR-LSQ-CMU), there is significant rotational error even after probing 250 points (see Table 3.2).

The estimated stiffness map from SCAR-LSQ-CMU (see Fig. 3-3(b)) looks very similar to the ground truth stiffness map (see Fig. 3-3(a)), unlike the stiffness map estimated by SCAR-IEKF-old (see Fig. 3-3(c)). This example demonstrates how the CMU overcomes the ambiguity described in section 3.2 and provides better estimates

Table 3.2: Comparison of registration results between SCAR-LSQ-CMU and SCAR-IEKF-old

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Actual	0	0	0	0	0	0	–
SCAR-LSQ-CMU	0	0	0	0	0	0	0
SCAR-IEKF-old	0.32	0.12	2.46	-4.45	-1.03	1.22	5.66

compared to SCAR-IEKF-old.

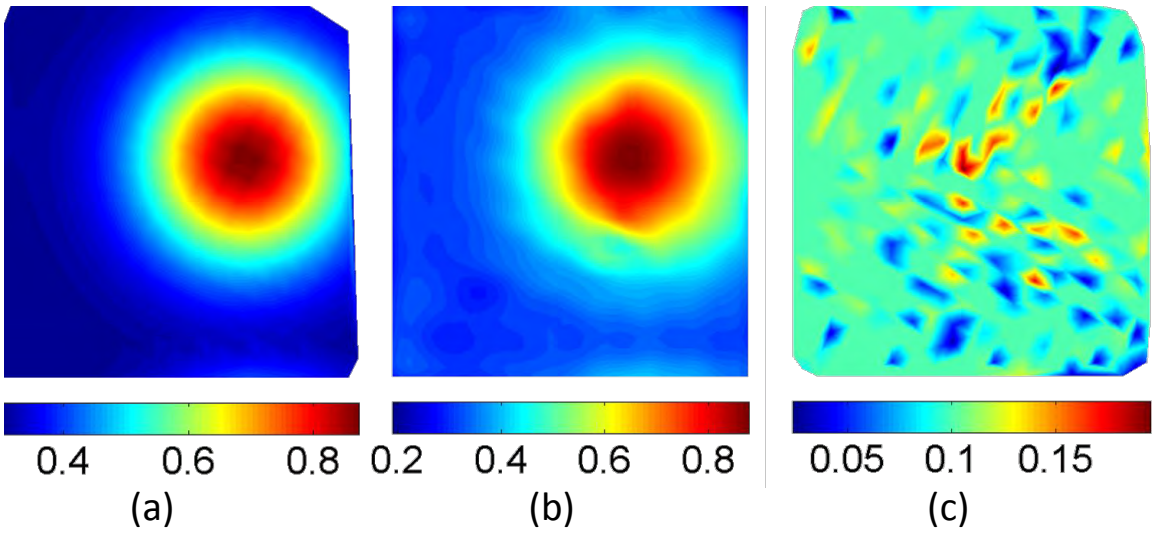


Figure 3-3: Stiffness in N/mm (a) Ground truth (b) Estimated by SCAR-LSQ-CMU (c) Estimated by SCAR-IEKF-old

3.3.2 Evaluation of Robustness to Sensor Noise

In order to test the robustness of the proposed algorithm to the presence of noise in the sensor measurements, we develop simulation data in which we artificially add noise to the measurements. While do not consider measurement outliers in the simulation experiments, we do observe outliers in the robot experiments. Any outliers present in the measurements would first affect the linear regression that is used to estimate the stiffness; which in turn would affect the registration. However, there exist several robust techniques in literature to deal with outliers in linear regression [126]. In practice we identify the k most likely outliers and use the remaining measurements

in the regression, which is a popular approach developed by Gentleman *et. al.* [127]. While we use $k = 10\%$ in this work, k can be chosen to be any other value depending on the prior knowledge of the outliers. We take the case of an organ whose shape is as shown in Fig. 3-6(a) with a synthetic ground truth stiffness map as shown in Fig. 3-4(a). The CAD model has 1311 triangle faces and is probed at 341 uniformly spaced points. At each palpated point, we record 10 measurements by probing along the normal up to a depth of 3mm in increments of 0.3mm. An artificial noise selected uniformly from $[0, 0.1]$mm and $[0, 0.1]$N is added to the sensed position and force respectively. Fig. 3-4(b) shows the stiffness map as estimated using SCAR-LSQ-CMU on this data. The stiffness map reveals the stiff features present in the ground truth. Following this we increase the noise in the sensed position, by selecting uniformly from $[0, 0.3]$mm. The stiffness estimation as shown in Fig. 3-4(c) demonstrates that SCAR-LSQ-CMU can reveal the stiff features even in the presence of high sensor noise.

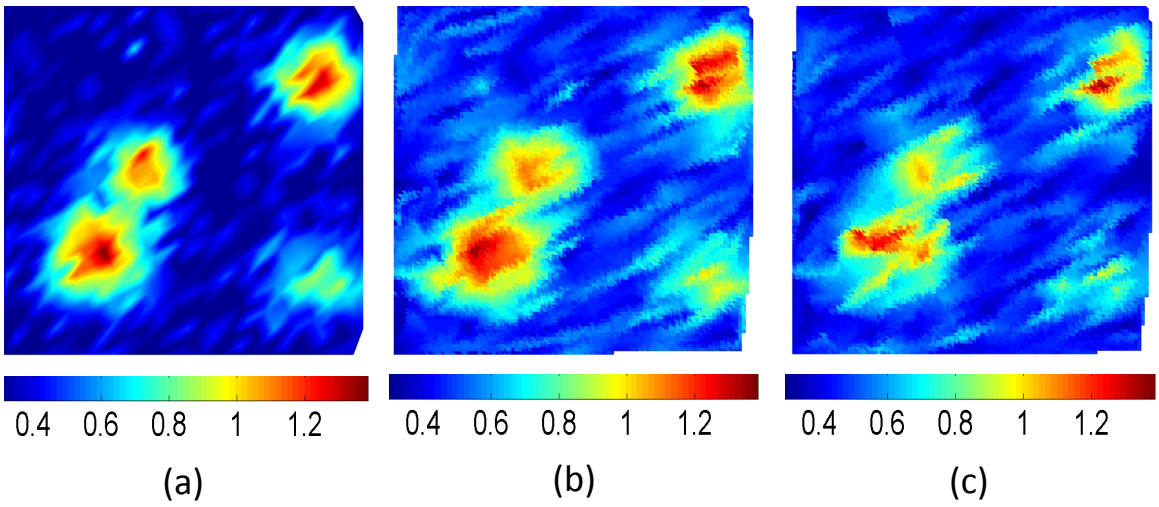


Figure 3-4: Stiffness in N/mm(a) Ground truth (b) Estimated under low sensor noise (c) Estimated under high sensor noise

Table 3.3 shows the comparison of the results as estimated by SCAR-LSQ-CMU with ICP [11], one of the most popular registration methods. Since ICP does not consider local deformations in its formulation, we create a modified formulation of the ICP to compensate for the local deformations (which we term deformation com-

Table 3.3: Registration results for different noise levels

Low noise level	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Actual	7	-12	15	-11.46	5.72	8.59	–
SCAR-LSQ-CMU	6.99	-11.94	14.99	-11.46	5.73	8.56	0.02
ICP	6.40	-11.32	17.39	-11.28	5.35	8.36	2.33
DICP	8.69	-13.63	17.12	-11.22	5.89	8.53	3.00
High noise level							
SCAR-LSQ-CMU	6.99	-11.96	15.00	-11.48	5.72	8.56	0.02
ICP	7.15	-13.14	17.14	-11.44	5.80	9.06	2.35
DICP	8.59	-14.05	17.42	-11.30	5.94	8.87	3.24

pensated ICP or deformation compensated ICP (DICP)), so that we can provide a fair comparison to SCAR-LSQ-CMU. In DICP we estimate the local deformation from the stiffness data and then displace the probed points along the sensed normal by the deformation depth to estimate the undeformed points[3]. Upon estimating the undeformed points, we use the original ICP to estimate the registration. In addition to finding the registration estimates, we also find the root mean square (RMS) error between the estimated positions and true positions, over all the probed points.

We assume the initial registration guess is $\boldsymbol{T}_0 = \boldsymbol{I}$, where $\boldsymbol{I}$ is an identity matrix. As expected, estimating local deformations results in SCAR-LSQ-CMU performing better than ICP and DICP in both the cases. DICP is affected by noise in sensed normal data resulting in a poor registration estimate, while SCAR-LSQ-CMU uses normal from the CAD model.

3.3.3 Evaluation of Robustness to Initial Registration Error

For the simulated example presented in Section 3.3.2 with lower sensor noise, we evaluate the registration estimates for two different initial registration errors. The initial registration error for Case 1 is lower than that for Case 2. From Table 3.4 we

[3]Since the publication of our work, there has been some recent work by Chalasani *et. al.* [111, 112], who use a similar approach to find the undeformed geometry of the organ surface. The difference being they use forces estimated from a GPR since they perform continuous palpation, while we use actual force measurements obtained from discrete probing.

observe than SCAR-LSQ-CMU estimates registration accurately even in the presence of high initial registration error.

Table 3.4: Evaluation of registration-robustness to initial conditions

Case1	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Initial	7.30	-12.10	15.61	-10.91	5.25	8.39	–
Actual	7	-12	15	-11.46	5.72	8.59	–
SCAR-LSQ-CMU	6.95	-11.94	14.99	-11.45	5.68	8.56	0.03
ICP	7.07	-12.25	16.26	-11.40	5.79	8.73	1.31
DICP	7.69	-11.86	16.21	-11.32	5.89	8.42	1.22
Case 2							
Initial	-13.31	-2.42	32.18	-30.76	-9.74	36.98	–
SCAR-LSQ-CMU	6.97	-11.91	15.01	-11.47	5.70	8.54	0.02
ICP	7.07	-12.25	16.26	-11.40	5.79	8.73	1.31
DICP	8.93	-11.93	16.51	-14.93	6.05	8.66	3.67

3.3.4 Experimental Validation

To evaluate our CMU algorithm we have used a custom designed Cartesian robot at Vanderbilt university with an open architecture controller [128] (see Fig. 3-5(a)). The robot end-effector was equipped with an ATI Nano43 F/T sensor. A target machine using Matlab Simulink® Real-Time operating system was used for the low level control at a control frequency of 1 KHz. For the probing and environment exploration, a hybrid motion/force controller was implemented as in Khatib [129]. The motion control was accomplished using proportional derivative inverse dynamics controller with a 5^{th} order polynomial trajectory generator. A proportional integral control law was used for force control. The force and motion reference commands were generated on a host machine which communicated with the low-level target machine using UDP. Post the publication of this work, our collaborators at Johns Hopkins university have developed a complementary situational awareness (CSA) software framework [130]. CSA is implemented as a component based framework, using the open source cisst libraries, developed at Johns Hopkins University [131, 132],

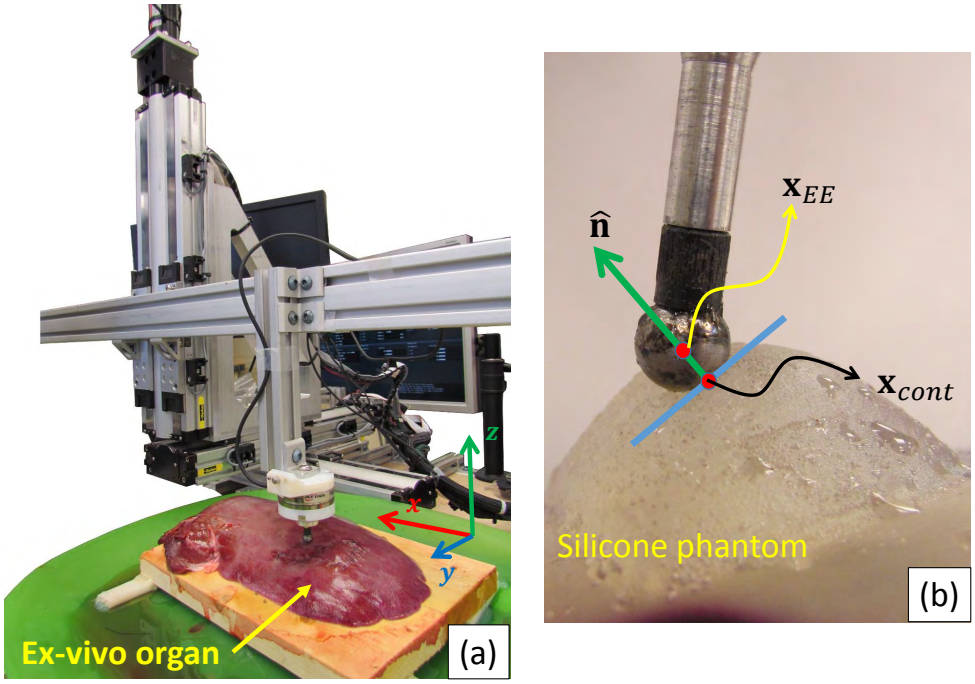


Figure 3-5: (a)Cartesian robot setup at Vanderbilt university used for experiments (b) Contact location and surface norm estimation

with support for the robot operating system (ROS) [133]. In the future we plan to use this framework to demonstrate our CMU approach on a daVinci research kit (dVRK) [134].

Robot Automatic Probing Procedures

Given a target region for exploration, the host machine generates a uniformly distributed grid map (uniform spacing in the $\hat{\mathbf{x}} - \hat{\mathbf{y}}$ plane of the robot) for probing locations. Given a particular reference probing location $\mathbf{x}_{\text{p}}$, the robot repeats the following steps to obtain the force-position measurements:

1. *Making high force contact:* The robot is first commanded to move to a desired position $\mathbf{x}_{\text{p}}$ and then to move along the Z direction until a force magnitude 0.5N is reached.

2. *Estimating surface norm:* The surface normal $\hat{\mathbf{n}}$ is computed as the direction of the sensed force: $\hat{\mathbf{n}} = \mathbf{f}_s / \|\mathbf{f}_s\|$. The location of the contact point on the surface can be computed as: $\mathbf{x}_{\text{cont}} = \mathbf{x}_{\text{EE}} - \hat{\mathbf{n}} r$ (see Fig. 3-5(b)).

3. *Finding low force surface contact point:* The robot first retrieves swiftly away from the surface along the direction of the estimated normal and then moves slowly towards the surface along the normal till the sensed force reaches a threshold.

4. *Probing and recording:* The force and position measurements are recorded as the robot moves up to a preset depth into the organ under position control.

Results of probing silicone model with stiff features

We use a silicone phantom organ with embedded stiff features as shown in Fig. 3-6(a) to test the performance of SCAR-LSQ-CMU. The phantom is probed at 1010 uniformly spaced points. The stiffness map as estimated from SCAR-LSQ-CMU reveals all four stiff features (see Fig. 3-6(b)). Table 3.5 shows the registration estimates from SCAR-LSQ-CMU, ICP and DICP. From Fig. 3-6(c) we notice that SCAR-LSQ-CMU has a lower RMS error than ICP and DICP after about 25 iterations. We also observe that the estimate converges with as few as 100 points for SCAR-LSQ-CMU as opposed to ICP and DICP (see Fig. 3-6(d)). This suggests that even though we have 1010 probed points, for an accurate registration estimation we only need to probe about 100 points randomly spread across the surface of the organ. It is important for the chosen points to be spread across the organ so that they can provide a reasonable estimate of the stiffness over the entire surface of the organ. Further, if the points chosen are not spread across, then the obtained measurements may not be descriptive enough to capture the geometry of the organ to be registered to. A Bayesian optimization-based procedure to optimally choose the points to probe that can quickly reveal the stiff inclusions, has been recently developed by Ayvali *et. al.* [2]. In the next section we show results for our CMU approach using measurements obtained from the Bayesian optimization approach.

Table 3.5: Registration results for experimental data

Silicone model	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Actual	5	-7	10	5.73	5.73	-8.59	–
SCAR-LSQ-CMU	4.96	-7.07	10.07	5.71	5.70	-8.50	0.11
ICP	4.96	-7.05	10.76	5.71	5.71	-8.51	0.80
DICP	6.67	-6.86	10.83	5.56	5.79	-8.79	1.87
Ex vivo organ							
Actual	7	8	-10	-5.73	-5.73	8.59	–
SCAR-LSQ-CMU	7.45	7.80	-9.81	-5.73	-5.48	8.69	0.28
ICP	5.56	7.39	-7.66	-0.43	-6.2	8.62	4.15
DICP	6.36	7.59	-9.66	-0.48	-5.84	8.52	3.49

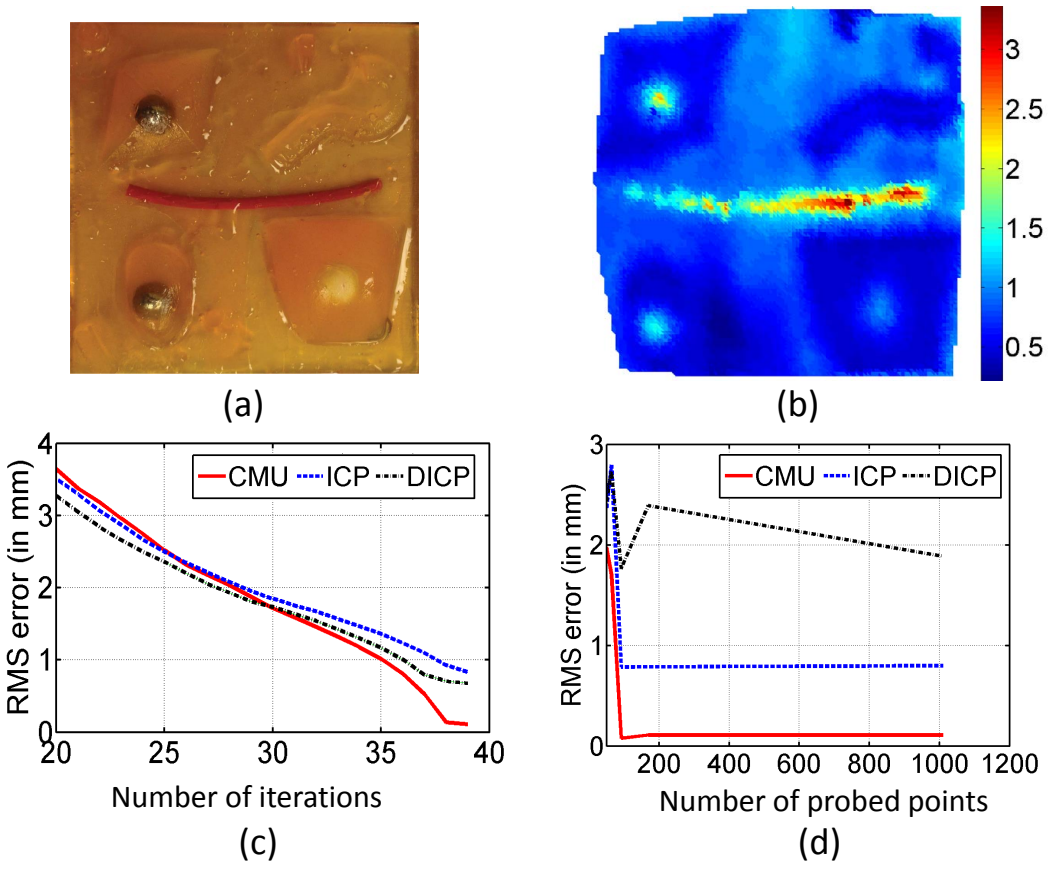


Figure 3-6: (a) Top view of the silicone organ (b) Stiffness map as estimated by SCAR-LSQ-CMU (Stiffness in N/mm) (c) Comparison of RMS error vs number of iterations (d) Comparison of RMS error vs number of probed points

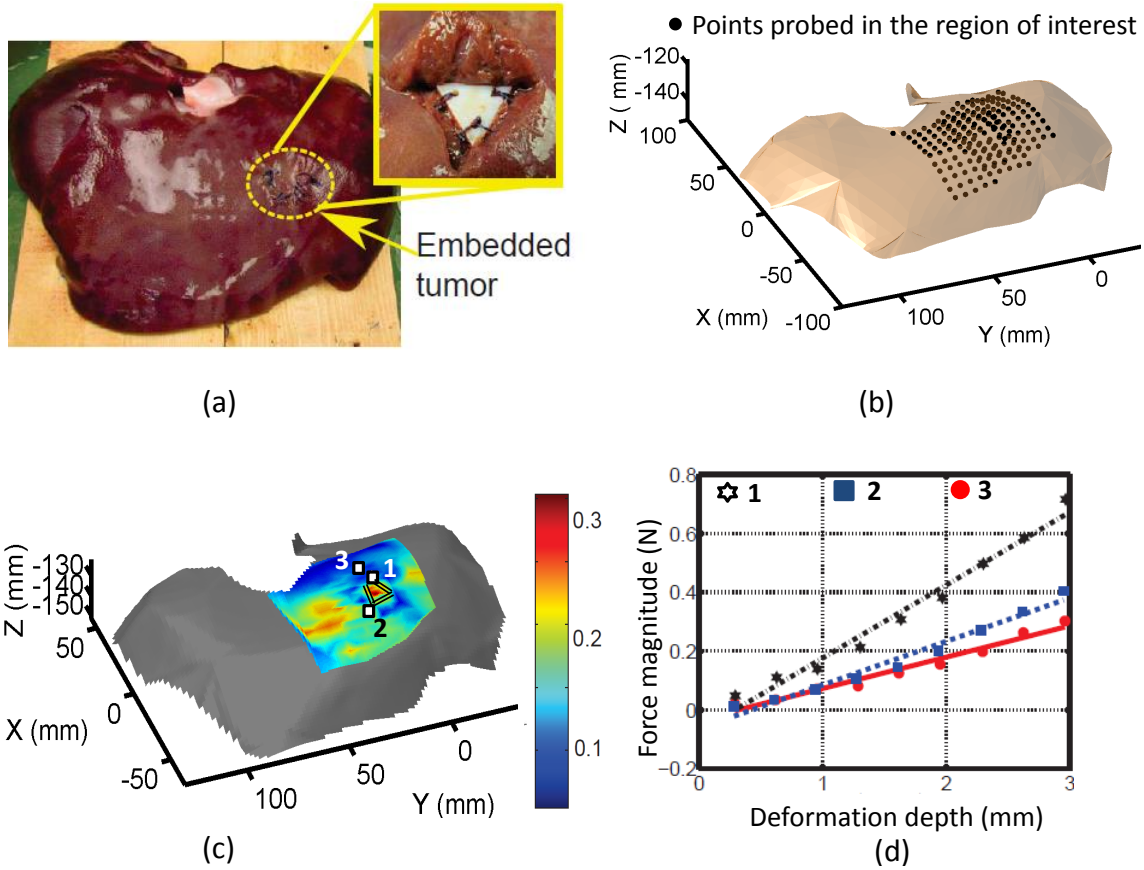


Figure 3-7: (a) An *ex vivo* porcine liver with artificially embedded tumor (b) Position of probed points on the surface of the organ (c) Stiffness map as estimated by SCAR-LSQ-CMU (Stiffness in N/mm)(d) Variation of applied force with deformation depth at three arbitrarily points chosen in (c)

Results of probing *ex vivo* organ

In order to validate the proposed work in a more realistic scenario, we palpate a porcine liver and use SCAR-LSQ-CMU for registration and stiffness estimation. A triangular plastic inclusion is artificially placed inside the organ and sutured as shown in Fig. 3-7(a). We palpate a region of interest on the liver at 196 equally spaced points. In order to obtain the location of the inclusion, we manually probe the organ with an optical marker and track the position using a Polaris Vicra optical tracking system. We then place an optical marker on the tool tip to compare the position of the tool-tip in the tracker's frame as well as in the robot's frame to obtain the ground-truth registration.

We obtain a stiffness map as shown in Fig. 3-7(c), which clearly shows the stiff inclusion. Fig. 3-7(d) shows linear variation of force with depth at three arbitrarily chosen locations on the surface, validating our assumption of a linear stiffness model. Table 3.5 shows the registration estimates for an initial registration guess of $\boldsymbol{T}_0 = \boldsymbol{I}$. The accuracy of registration as required for clinical applications generally depends on the size of the smallest tumor that needs to be removed and the resection margin [135, 136]. In this example, the size of the tumor is 2cm and the registration accuracy obtained is 0.28mm. Hence we can safely say that the obtained registration accuracy is sufficient to discern the tumor.

3.3.5 Results for Bayesian optimization guided probing

In the results demonstrated thus far, we considered exhaustively probing the entire surface of the organ. Since exhaustive palpation can be very time consuming, the SCAR-LSQ-CMU approach was used along with the Bayesian optimization approach introduced by Ayvali *et. al.* [2] to simultaneously register and estimate the stiffness distribution with minimal number of probings (see Fig. 3-9). The approach of Ayvali *et. al.* [2] uses Gaussian processes to model the stiffness distribution and Bayesian optimization to direct where to sample next for maximum information gain. Instead of finding the next point to probe, one may alternatively find the best continuous

trajectory to probe. Continuous trajectories can provide a greater density of measurements compared to discrete probing. Discrete probing, however, has the benefit of providing a more accurate assessment of tissue stiffness at the probed point. On the other hand, in continuous palpation, each force measurement corresponds to a different undeformed surface location, which makes the tissue stiffness estimation more challenging. Chalasani *et. al.* [111] developed a Gaussian process-based approach to estimate the stiffness and the underformed surface point from force and displacement data obtained during continuous palpation. In the future, we plan to use the SCAR-LSQ-CMU approach along with the continuous palpation approach of Chalasani *et. al.* [111]. Post the publication of our work, there have been other continuous palpation approaches using Bayesian optimization by Chalasani *et. al.* [112], ergodic coverage by Ayvali *et. al.* [137] and active area search by Salman *et. al.* [138].

For the rest of this section, we use the discrete probing approach of Ayvali *et. al.* [2] to provide inputs to the SCAR-LSQ-CMU approach. We start with an initial set of 19 samples that are uniformly distributed over a grid on the surface of the organ and terminate the palpation after 100 points. Fig. 3-8(a) shows where we think the position of all the probed points are based on the initial registration guess and their registered position estimated by SCAR-LSQ-CMU. The registered position of the probed points (deformed points) lie below the surface of the CAD model as expected; validating that the registration estimate is accurate. Fig. 3-8(b) and (c) show the ground truth stiffness map and the predicted stiffness map, respectively. The predicted stiffness map captures the stiff features present in the ground truth stiffness map. On Matlab, predicting the stiffness distribution and determining the next point to palpate takes 0.11s on average while updating the registration takes 1.2s on average for 100 probed points. Table 3.6 shows the registration results for this experiment.

The required clinical accuracy of registration is application dependent. For tumor localization, the registration accuracy can be dictated by the size of the smallest tumor that needs to be resected or ablated [135]. For example, the smallest tumor

Table 3.6: Registration results for CMU approach with Bayesian optimization guided probing

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Actual	5	10	-15	11.46	-11.46	5.73	–
Guess	0	0	0	0	0	0	–
SCAR-LSQ-CMU	4.56	9.24	-14.97	12.58	-11.27	5.69	1.14

size detected in livers is 1cm on average [139]. Hence, we conclude that the registration accuracy we achieve is well above the requirements.

3.3.6 Evaluation in Presence of Stiffness Priors

A rotationally symmetric object has multiple solutions for rotation and/or translation, resulting in an ambiguity in registration [17]. In other applications, this ambiguity is usually resolved by introducing an additional dimension such as surface texture [140], surface reflectance [141], etc.

In order to develop a formulation for registration that works reliably for any organ geometry, we extend the CMU formulation by using a stiffness prior in addition to the geometric prior for resolving the ambiguity in registration.

A prior stiffness map can be generated using elastography, physics based simulations or other complementary methods. In this work we generate the prior stiffness using a physics based simulation that assumes a linear stiffness model. The stiffness values are normalized and classified into two discrete levels, high and low stiffness, using Otsu method [142].

The only modification to the CMU formulation happens in the correspondence step. In order to ensure that a point corresponding to a high stiffness region on the model-frame is mapped to a point with high stiffness in the robots frame, we normalize and classify the estimated stiffness map using [142] (Fig. 3-10(c) was generated from Fig. 3-10(a)). We choose the point on the preoperative model that is closest and also has the same discrete stiffness level in the prior stiffness map.

The stiffness map estimated by SCAR-LSQ-CMU using stiffness prior is shown in

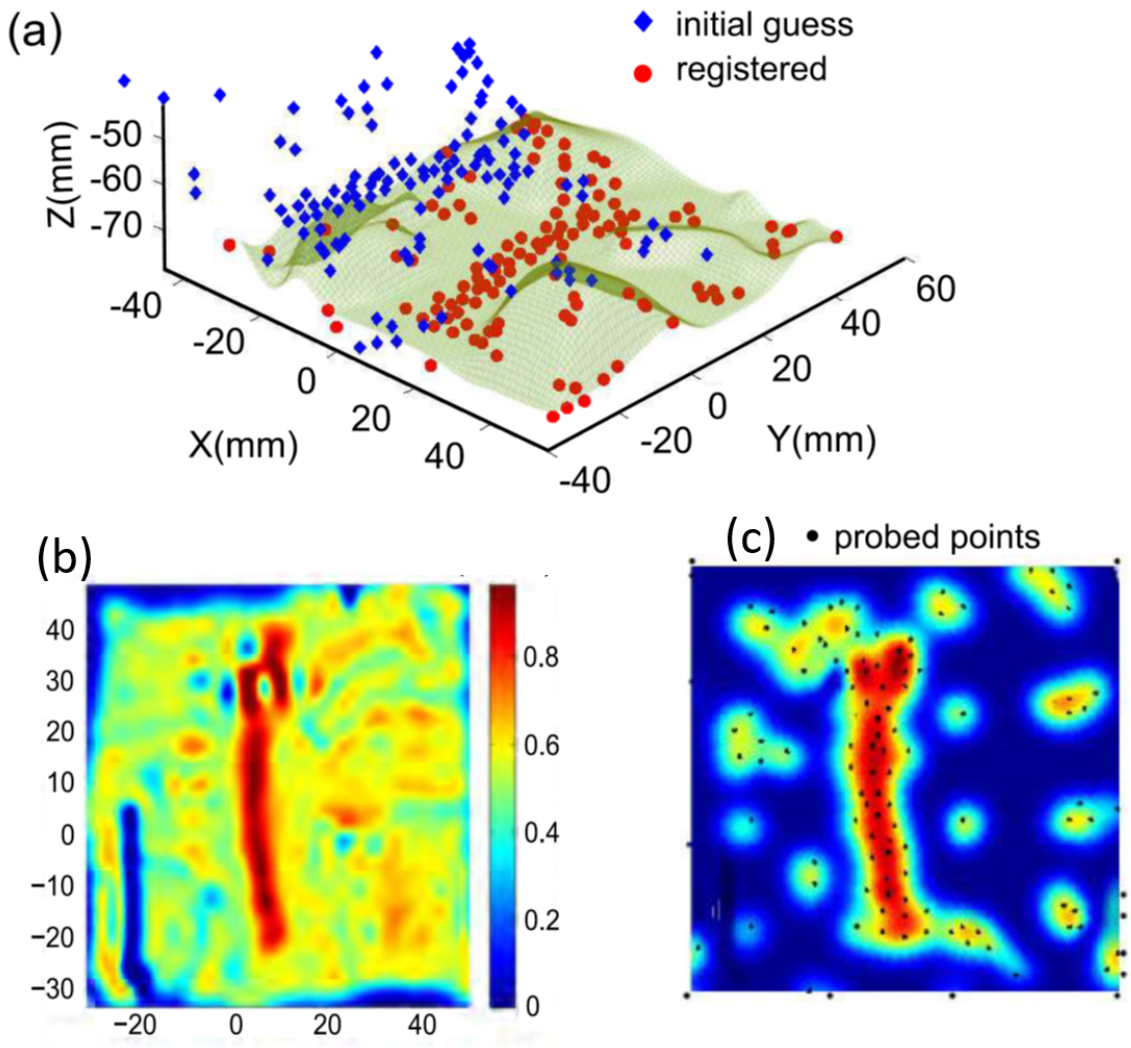


Figure 3-8: (a) Registration results for Example 3, (b) Ground truth stiffness map (c) Estimated map for Example 3 with 119 probed points. The algorithm starts with an initial grid of 19 points: 4 corners and 15 uniformly spaced points.

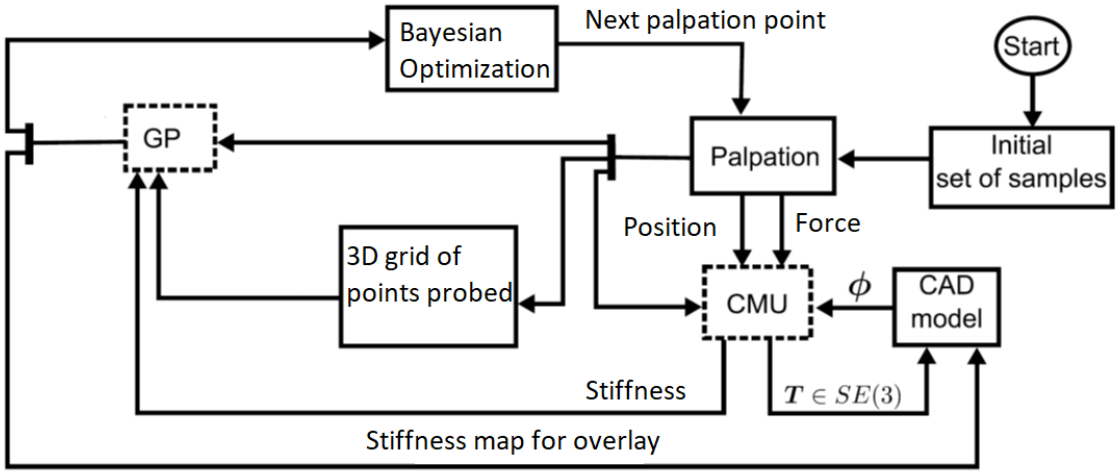


Figure 3-9: Block diagram description of the Bayesian optimization guided probing (Courtesy [2]).

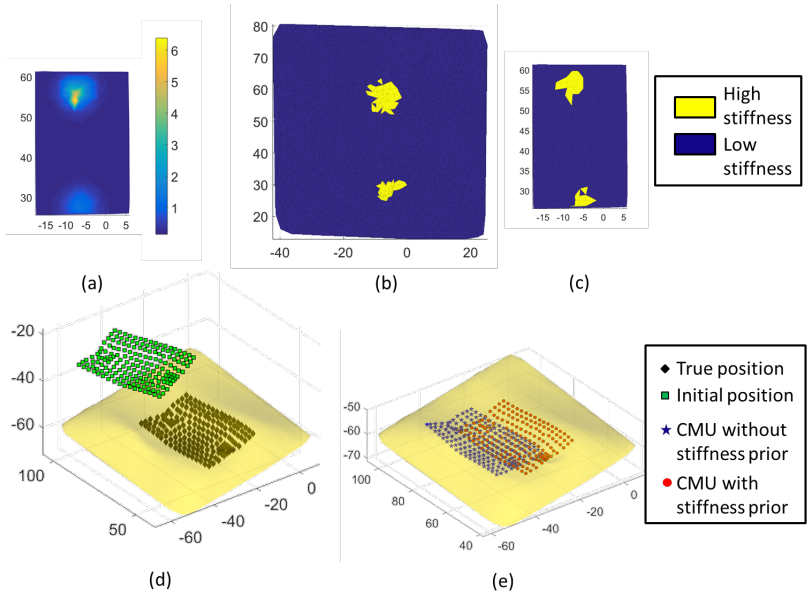


Figure 3-10: (a) Estimated stiffness map (stiffness in N/mm). (b) and (c) Prior stiffness map and estimated stiffness map respectively, normalized and stiffness values classified to high and low stiffnesss levels. (d) Initial and true location of probed points. (e) Estimated location probed points.

Fig. 3-10(a). Note that the two stiff inclusions are clearly visible in the stiffness map. In Fig. 3-10(d), black-diamond markers show the 180 points that were probed in the region of interest. Green-square markers show the initial guess for the location of the probed points. In Fig. 3-10(e), blue-star markers show the position as estimated by SCAR-LSQ-CMU without using a stiffness prior. Red-circular markers show the position estimated by SCAR-LSQ-CMU using the stiffness prior.

Table 3.7 shows the RMS error for ICP and SCAR-LSQ-CMU with and without stiffness prior, for a representative example. SCAR-LSQ-CMU with stiffness prior estimates the registration parameters very accurately and the RMS error is within clinical requirements [135]. In this work, we use the value of the stiffness level to filter out incorrect correspondences. It is however worth noting that we could also associate weights to the matching based on proximity to the model and the stiffness level. Such an approach could potentially perform better when the prior stiffness map does not contain location of all stiff inclusions or when the location of stiff inclusions is not accurate.

Table 3.7: Registration Evaluation in Presence of Stiffness Prior

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Initial	0	0	0	0	0	0	–
Actual	-20	15	15	-10	11.46	-8.59	5.73 –
SCAR-LSQ-CMU	-16.4	19.9	-14.8	15.45	5.84	8.16	7.74
SCAR-LSQ-CMU with stiffness prior	-21	16.73	-9.1	11.28	-8.6	5.23	2.19
ICP	-18.9	20.5	-15.4	16.21	7.35	6.5	7.77

3.4 Conclusion

We have presented a new and robust formulation that uses mechanical palpation to simultaneously estimate the stiffness distribution and register preoperative models to visible anatomy. We believe the proposed model update is not a replacement but instead complementary to existing intraoperative registration methods. The performance of the new update method to several initial conditions, different geometries,

stiffness profiles as well as sensitivity to sensor noise was evaluated and its robustness was demonstrated by a number of examples. We also introduce deformation compensated ICP (DICP) as an improvement over ICP, to provide a more fair comparison to SCAR-LSQ-CMU. We show that SCAR-LSQ-CMU performs better than SCAR-IEKF-old, DICP and ICP. We also show that the performance of CMU can be further improved by using stiffness priors.

While we presented an implementation of this update model with a least squares optimizer in this paper, it must be noted that the update model can be used with other optimizers as well as filtering approaches such as [78, 5], which forms part of future work. In this work, we use a simple experimental setup for evaluating our method, while avoiding additional sources of error such as robot deflection and positional errors typically seen in existing surgical systems. However, we plan to deploy our algorithm on research platforms that address key issues of surgical access constraints during MIS and offer force sensing capabilities.

3.5 Contribution and Discussions

The contributions from this chapter include:

1. Development of a complementary model update that simultaneously estimates stiffness map and registration.

2. Using a function of mini-batch of measurements as a pseudo-measurement for updating the state.

Fig. 3-11 shows a schematic for state updates using a hash of measurements. From a stream of measurements, we form pseudo-measurements by first separating red measurements from green and evaluating a function of each category of measurements to form red and green pseudo-measurements (See Fig. 3-11). These pseudo-measurements along with the prior are then used to update the state. The novelty of this approach is – (i) Not every measurement is directly used to update the state, but instead a function of the measurements is used (for example, refer to the collection

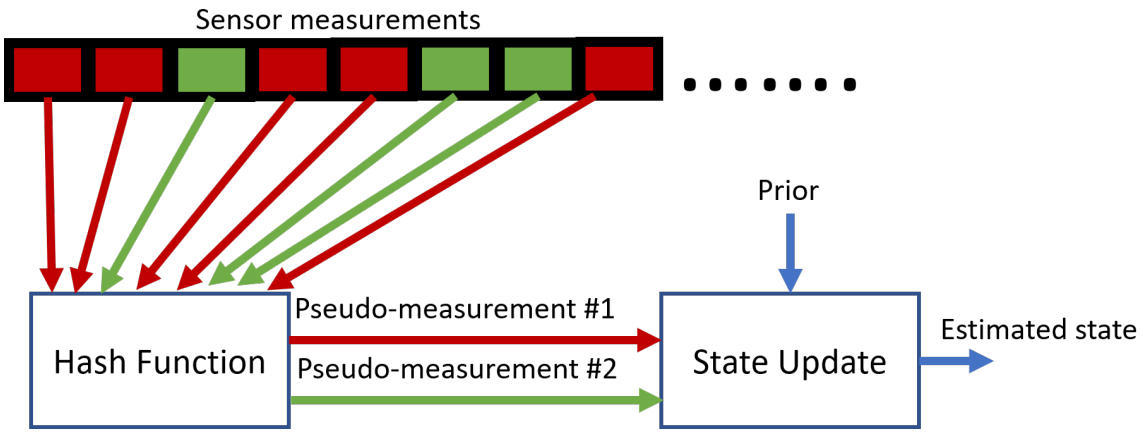


Figure 3-11: Schematic showing state update using a hash of measurements.

step of CMU in Sec. 3.2), (ii) we categorize measurements to compatible groups and then evaluate a function of those measurements to form pseudo-measurements, in contrast to standard pseudo-measurement-based updates where a pseudo-measurement is constructed by evaluating a function of every measurement (for example refer to the Stiffness estimation step of CMU in Sec. 3.2).

In the case of SCAR, using the 'hash' update helps resolve the ambiguity that existed in the prior approach by Sanan *et. al.* [78]. We demonstrate in the next chapter, Chapter 4, that the hash-update helps derive a linear model for pose update resulting in more accurate and fast estimates of pose.

3.6 Published Work

Material from this chapter has appeared in the following publications

1. R Arun Srivatsan, Elif Ayvali, Long Wang, Rajarshi Roy, Nabil Simaan and Howie Choset, "Complementary Model Update: A Method for Simultaneous Registration and Stiffness Mapping in Flexible Environments", In the proceedings of the International Conference on Robotics and Automation, Stockholm, Sweden, May 2016, pp. 924-930.

2. R. Arun Srivatsan, Long Wang, Elif Ayvali, Nabil Simaan, and Howie Choset,

"Simultaneous Registration and Stiffness mapping of a Flexible Environment using Stiffness and Geometric Prior, in the proceedings of the Hamlyn symposium on Medical Robotics, UK, June 2016.

3. Elif Ayvali, R Arun Srivatsan, Long Wang, Rajarshi Roy, Nabil Simaan and Howie Choset, "Using Bayesian optimization to guide probing of a flexible environment for simultaneous registration and stiffness mapping", In the proceedings of the International Conference on Robotics and Automation, Stockholm, Sweden, May 2016, pp. 931-936.

Chapter 4

Dual Quaternion Filter for Pose Estimation

Many applications in robotics such as registration, object tracking, sensor-calibration, etc. use Kalman filters to estimate a time invariant pose by locally linearizing a non-linear measurement model. In this work, we use a dual quaternion to represent the pose and use pairs of measurements simultaneously to rewrite the measurement model in a truly linear form with state dependent measurement noise. The intuition to form pseudo-measurements using a pair of measurements simultaneously, was obtained from our work on SCAR-LSQ-CMU which is described in Chapter 3.

Dual quaternions provide a means to compactly combine both rotation and translation in an unambiguous and singularity free manner [94]. While dual-quaternions have been used with IEKF to estimate pose, the update model was non-linear [64]. Non-linear update models can be highly sensitive to initial estimation errors, and can be computationally expensive. As a result in this work, we focus on deriving a linear update model to estimate pose. Chaukron *et. al.* [143] come closest to our work in terms of formulating a linear update model, but they only estimate the $SO(3)$ element. In this work we use multiple sensor measurements simultaneously to rearrange the originally nonlinear update model into a linear form. To the best of our knowledge, this is the first attempt to derive a linear update model for estimating time invariant pose using a Kalman filter.

The linear measurement model comes at the cost of state dependent measurement uncertainty. Measurement uncertainty is typically state independent and can be obtained based on the physical characteristics of the sensor and/or the measurement process. However, in case of state dependence, there is an additional burden of estimating the measurement uncertainty after each state update. State dependent measurement uncertainties have been used in systems for satellite tracking [144] and robot navigation [145]. We use an approach similar to Shuster *et. al.* [144] and Chaukroun *et. al.* [143] to formulate the expressions for the state dependent measurement uncertainties. It should be noted that the measurement uncertainties have a linear dependence on their state vector, which allows for derivation of exact expressions of uncertainties [146].

We consider two broad classes of applications in this section based on the type of measurements used to estimate the pose: 1) those that use position measurements such as registration from medical imaging [5], object tracking using laser range scanners [147], etc. and 2) those that use pose (position and orientation) measurements such as sensor calibration using inertial measurement units [148], hand-eye calibration using stereo vision [31], etc . The linear measurement models and state-dependent uncertainties are derived for both of these cases. We develop a dual quaternion-based filter (DQF) for pose estimation in this section and compare the results with non-linear filtering variants. We evaluate the formulation through simulations and experiments for two applications: registration and sensor calibration. DQF produces more accurate and fast estimates even in the presence of high initial errors.

4.1 Related Work

Estimation of $SE(3)$ elements has been of interest for a long time in robotics literature. Horn [4] and Arun *et. al.* [17] developed methods for least squares estimation of $SE(3)$ elements. While Horn uses quaternions to parameterize the orientation, Arun *et. al.* use an orthogonal matrix to parameterize the rotation. While the methods of both Horn and Arun *et. al.* first estimate the orientation and then estimate the translation,

Walker *et. al.* [61] used dual quaternions and developed an approach to simultaneously estimate rotation and translation. Wang *et. al.* [100] recently improved the work of Walker *et. al.* but estimating scale in addition to the pose. Besl and McKay [11] developed the popular ICP approach for finding the pose between a set of points in one frame and a model or another set of points in another frame. The ICP iteratively estimates the correspondence between the points and minimizes an objective function to find a locally optimal pose.

Park *et. al.* [149] use Lie algebra elements and a least squares optimization approach to estimate the orientation and translation in a decoupled form. Chen [150] used a screw-theoretic approach to provide an analysis on the conditions under which a solution for the hand-eye calibration problem can be uniquely found. Daniilidis *et. al.* [99, 32] developed upon the result of Chen by using dual quaternions to simultaneously finds the best rotation and translation using a singular value decomposition. More recently, Zhang *et. al.* [151] have developed an iterative method for online estimation of pose in hand-eye calibration applications, using dual quaternions.

In the presence of noisy measurements, deterministic optimization methods have been observed to perform poorly [6]. On the other hand, probabilistic estimation techniques such as Kalman filters are effective at handling noisy measurements and producing accurate estimates of the state and associated uncertainty [51]. Several researchers have noted that filters used for pose estimation have non-linear update models [49], and hence variants of the Kalman filter have been introduced to handle this non-linearity. The EKF and UKF have been used to estimate $SE(3)$ elements for satellite orientation [144], manipulation [152], registration [6, 5] and sensor calibration [10]. EKF-based filters perform first-order linear approximations of the non-linear update models and produce estimates which are known to diverge in the presence of high initial estimation errors sa observed by Chaukroun *et. al.* [143]. UKF based methods do not linearize the models but instead require evaluation at multiple specially chosen points (called sigma points), which can be expensive for a high-dimensional system such as $SE(3)$. In addition UKF-based methods require tuning of multiple parameters, which is not intuitive.

Prior work has also looked at several parameterizations of $SE(3)$ that would improve the performance of the filters. In the work of Hauberg *et. al.* [49] the state variables are confined over a known Riemannian manifold and a UKF is used to estimate the $SE(3)$ element. Exponential coordinates were used with an EKF in Chapter 5. Janabi and Marey [153] use Euler angles and Cartesian coordinates to parameterize the pose and developed an iterative adaptive extended Kalman filter for pose estimation. All the above methods involve highly non-linear update models with trigonometric terms in them. Bar *et. al.* [154] use unit-quaternions to parametrize the rotation component of $SE(3)$ and an EKF to estimate the pose. Laviola *et. al.* [63] also use unit-quaternions but estimate the pose using a UKF. Dual quaternions with an IEKF has been used by Goddard *et. al.* [64]. Goddard *et. al.* however, do not use the dual quaternion as the state vector. They instead include Cartesian coordinates and a unit quaternion in their state vector and calculate the dual component from the estimated translation. Unlike Goddard *et. al.*, Filipe *et. al.* [101] include all the terms of a unit dual quaternion in their state vector and use an EKF for state update. They impose the inherent constraints of the unit dual quaternion after each state update.

In this work, we use dual-quaternions to represent the $SE(3)$ element; using multiple simultaneous measurements, we derive a linear update model which can be used with a Kalman filter without the need for linearization.

4.2 Problem Formulation

Most applications that estimate time invariant pose can be broadly divided into two cases: Case I, ones that use position measurements and Case II, that use pose measurements for updating the state. The measurement model for both these cases are non-linear and algebraically very different. Dual quaternions provides the means to rewrite the measurement models for both these cases in a linear form. The rest of this chapter deals with the derivation of measurement models for the two cases and the corresponding uncertainties.

4.2.1 Measurement Model for Position Measurements

Systems that use position-measurements for model update have the following general form

$$\boldsymbol{a} = \boldsymbol{R}\boldsymbol{b} + \boldsymbol{t}, \tag{4.1}$$

where $\boldsymbol{a}$ is a noise free sensor measurement, $\boldsymbol{R} \in SO(3)$ is the rotation matrix, $\boldsymbol{b} \in \mathbb{R}^3$ is the point to be transformed and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector. In an application such as rigid registration to a geometric model, $\boldsymbol{a}$ is the sensed location of points and $\boldsymbol{b}$ is the corresponding point on the geometric model of the object. Eq. 4.1 can be rewritten using dual-quaternions from Eq. 2.21 as shown

$$\hat{\boldsymbol{a}} = \hat{\boldsymbol{d}} \otimes \hat{\boldsymbol{b}} \otimes \hat{\boldsymbol{d}}^{3*}, \tag{4.2}$$

where $\hat{\boldsymbol{d}}$ is as defined in Eq. 2.19. Applying Lemma 2.4.1, Eq. 4.2 can be rewritten as

$$\hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{d}}^{1*} = \hat{\boldsymbol{d}} \otimes \hat{\boldsymbol{b}}. \tag{4.3}$$

Let us consider the case of a pair of measurements $\boldsymbol{a}_i$, $i = 1, 2$. From Eq. 4.3, we have

$$\begin{aligned}
&\hat{\boldsymbol{a}}_i \otimes \hat{\boldsymbol{d}}^{1*} = \hat{\boldsymbol{d}} \otimes \hat{\boldsymbol{b}}_i, \\
\Rightarrow &(1 + \epsilon\widetilde{\boldsymbol{a}}_1) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (1 + \epsilon\widetilde{\boldsymbol{b}}_1), \quad \text{and} && (4.4) \\
&(1 + \epsilon\widetilde{\boldsymbol{a}}_2) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (1 + \epsilon\widetilde{\boldsymbol{b}}_2) && (4.5)
\end{aligned}$$

Subtracting Eq. 4.5 from Eq. 4.4, we obtain

$$\begin{aligned}
&(\epsilon\,(\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2)) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes \left(\epsilon\left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right)\right) \\
&\Rightarrow \epsilon\,((\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r) = \epsilon\left(\widetilde{\boldsymbol{q}}_r \odot \left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right)\right) \\
&\Rightarrow (\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \left(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2\right) = \tilde{\boldsymbol{0}}. && (4.6)
\end{aligned}$$

Note that Eq. 4.6 does not have $\widetilde{\boldsymbol{q}}_d$ and contains only the rotation quaternion. Using the quaternion multiplication described in Eq. 2.11, Eq. 4.6 can be rewritten in the following form

$$\boldsymbol{H}\widetilde{\boldsymbol{q}}_r = \tilde{\mathbf{0}}, \quad \text{where} \tag{4.7}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & -(\boldsymbol{a}_1 - \boldsymbol{a}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2)^T \\ (\boldsymbol{a}_1 - \boldsymbol{a}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2) & (\boldsymbol{a}_1 - \boldsymbol{a}_2 + \boldsymbol{b}_1 - \boldsymbol{b}_2)^\times \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{4.8}$$

The rotation quaternion $\widetilde{\boldsymbol{q}}_r$ lies in the null space of $\boldsymbol{H}$. In order to estimate $\widetilde{\boldsymbol{q}}_r$ we use a Kalman filter whose state vector is $\widetilde{\boldsymbol{q}}_r$. For this filter, the pseudo-measurement model is

$$\boldsymbol{h} = \boldsymbol{H}\widetilde{\boldsymbol{q}}_r, \quad \text{where } \boldsymbol{h} \in \mathbb{R}^4. \tag{4.9}$$

We enforce the pseudo-measurement $\boldsymbol{h} = \mathbf{0}$. The measurement in Eq. 4.9 is called "pseudo-measurement" because $\boldsymbol{h}$ does not represent a true measurement (refer to Sec. 7.1 for a discussion on pseudo-measurement models). The pseudo-measurement model, is dependent on the state $\widetilde{\boldsymbol{q}}_r$, and sensor measurements $\widetilde{\boldsymbol{a}}_i$ and $\widetilde{\boldsymbol{b}}_i$ all of which have associated uncertainties. In section 4.2.3, we discuss the procedure to compute the uncertainty in the pseudo-measurement. Subsequently in section 4.3, we describe the equations of the Kalman filter that estimates $\widetilde{\boldsymbol{q}}_r$ using the linear measurement model.

After estimating $\widetilde{\boldsymbol{q}}_r$ using a Kalman filter, we need to estimate $\widetilde{\boldsymbol{q}}_t$. Adding the Eq. 4.4 and Eq. 4.5 we have

$$\begin{aligned} &(2 + \epsilon(\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2)) \otimes (\widetilde{\boldsymbol{q}}_r - \epsilon\widetilde{\boldsymbol{q}}_d) = (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes \left(2 + \epsilon(\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2)\right), \\ \Rightarrow &2\widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r = (\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot (\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2), \\ \Rightarrow &\widetilde{\boldsymbol{q}}_t = \frac{\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2}{2} - \widetilde{\boldsymbol{q}}_r \odot \frac{\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*. \end{aligned} \tag{4.10}$$

Thus, Eq. 4.10 computes $\widetilde{\boldsymbol{q}}_t$ directly using the estimated value of $\widetilde{\boldsymbol{q}}_r$ without the

need for a Kalman filter. This is a helpful byproduct of using multiple measurements simultaneously in Eq. 4.4. Since the scalar part of $\widetilde{\boldsymbol{q}}_t$ is 0 and vector part is $\boldsymbol{t}$, we can rewrite Eq. 4.10 in the following vector form

$$\boldsymbol{t} = \frac{\boldsymbol{a}_1 + \boldsymbol{a}_2}{2} - \boldsymbol{R}_{\widetilde{\boldsymbol{q}}_r} \left(\frac{\boldsymbol{b}_1 + \boldsymbol{b}_2}{2} \right), \tag{4.11}$$

where $\boldsymbol{R}_{\widetilde{\boldsymbol{q}}_r}$ is the rotation matrix formed using the quaternion $\widetilde{\boldsymbol{q}}_r$. Section 4.2.3 describes the uncertainty associated with $\boldsymbol{t}$.

4.2.2 Measurement Model for Pose Measurements

Systems that use pose-measurements for model update typically have the following general form [149]

$$\boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B} = \boldsymbol{0}, \tag{4.12}$$

where $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{B} \in SE(3)$. $\boldsymbol{A}$ and $\boldsymbol{B}$ are noisy pose-measurements and $\boldsymbol{X}$ is the desired transformation to be estimated.

A Kalman filter used to estimate $\boldsymbol{X}$ such as in [10], would have a pseudo-measurement model of the form, $\boldsymbol{h} = \boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}$, $\boldsymbol{h} \in \mathbb{R}^{3\times 3}$. One again we enforce the pseudo-measurement $\boldsymbol{h} = \boldsymbol{0}$. A UKF with a state matrix instead of state vector can directly handle measurement models in matrix forms [49]. The pseudo-measurements can also be converted to a vector form as shown in [10] and then estimated using a UKF. Using dual quaternions we rewrite Eq. 4.12 in an alternate form, which would ultimately result in a linear pseudo-measurement, thus allowing us to use a linear Kalman filter for state estimation.

Let $\hat{\boldsymbol{a}}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{b}}$ be the dual quaternions corresponding to $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{B}$ respectively. Eq. 4.12 can be rewritten as

$$\hat{\boldsymbol{a}} \otimes \hat{\boldsymbol{x}} - \hat{\boldsymbol{x}} \otimes \hat{\boldsymbol{b}} = \hat{\boldsymbol{0}}. \tag{4.13}$$

Using Eq. 2.19, Eq. 4.13 can be written as

$$\begin{aligned}\hat{\mathbf{0}} =&(\widetilde{\boldsymbol{a}}_r + \epsilon\widetilde{\boldsymbol{a}}_d) \otimes (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) - (\widetilde{\boldsymbol{q}}_r + \epsilon\widetilde{\boldsymbol{q}}_d) \otimes (\widetilde{\boldsymbol{b}}_r + \epsilon\widetilde{\boldsymbol{b}}_d),\\ =&\left(\widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r\right) +\\ &\epsilon\left(\widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r + \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_d - \widetilde{\boldsymbol{q}}_d \odot \widetilde{\boldsymbol{b}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d\right). \end{aligned} \tag{4.14}$$

Hence we have

$$\widetilde{\mathbf{0}} = \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \tag{4.15}$$

$$\widetilde{\mathbf{0}} = \widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r + \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_d - \widetilde{\boldsymbol{q}}_d \odot \widetilde{\boldsymbol{b}}_r - \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d. \tag{4.16}$$

Eq. 4.15 has a form very similar to Eq. 4.6, with the only difference being that the scalar parts of $\widetilde{\boldsymbol{a}}_r, \widetilde{\boldsymbol{b}}_r$ are not 0. If $\widetilde{\boldsymbol{a}}_r = a_0 + \boldsymbol{a}_r$ and $\widetilde{\boldsymbol{b}}_r = b_0 + \boldsymbol{b}_r$, using Eq. 2.11 we rewrite Eq. 4.15 as

$$\boldsymbol{H}_r\widetilde{\boldsymbol{q}}_r = \mathbf{0}, \text{ where} \tag{4.17}$$

$$\boldsymbol{H}_r = \begin{bmatrix} a_0 - b_0 & -(\boldsymbol{a}_r - \boldsymbol{b}_r)^T \\ \boldsymbol{a}_r - \boldsymbol{b}_r & (\boldsymbol{a}_r + \boldsymbol{b}_r)^\times + (a_0 - b_0)\,\boldsymbol{I}_3 \end{bmatrix} \in \mathbb{R}^{4\times 4}. \tag{4.18}$$

The pseudo-measurement model is

$$\boldsymbol{h}_r = \boldsymbol{H}_r\widetilde{\boldsymbol{q}}_r, \tag{4.19}$$

and the pseudo-measurement $\boldsymbol{h}_r = \mathbf{0}$, where $\boldsymbol{h}_r \in \mathbb{R}^4$. The uncertainty associated with $\boldsymbol{h}_r$ is derived in section 4.2.3.

Similar to section 4.2.1, we use the estimated value of $\widetilde{\boldsymbol{q}}_r$ to estimate $\boldsymbol{t}$. Using Eq. 2.20, Eq. 4.16 can be rewritten as

$$\widetilde{\mathbf{0}} = \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r + \widetilde{\boldsymbol{\sigma}}_1, \tag{4.20}$$

where $\widetilde{\boldsymbol{\sigma}}_1 = 2\widetilde{\boldsymbol{a}}_d \odot \widetilde{\boldsymbol{q}}_r - 2\widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_d$. Multiplying both sides of Eq. 4.20 with $\widetilde{\boldsymbol{q}}_r^*$, we

obtain:

$$\begin{aligned}\widetilde{\mathbf{0}} &= \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \odot \widetilde{\boldsymbol{q}}_r^* + \widetilde{\boldsymbol{\sigma}}_1 \odot \widetilde{\boldsymbol{q}}_r^*, \\ &= \widetilde{\boldsymbol{a}}_r \odot \widetilde{\boldsymbol{q}}_t - \widetilde{\boldsymbol{q}}_t \odot \widetilde{\boldsymbol{\sigma}}_2 + \widetilde{\boldsymbol{\sigma}}_3, \end{aligned} \tag{4.21}$$

where $\widetilde{\boldsymbol{\sigma}}_2 = \widetilde{\boldsymbol{q}}_r \odot \widetilde{\boldsymbol{b}}_r \odot \widetilde{\boldsymbol{q}}_r^*$ and $\widetilde{\boldsymbol{\sigma}}_3 = \widetilde{\boldsymbol{\sigma}}_1 \odot \widetilde{\boldsymbol{q}}_r^*$. The structure of Eq. 4.21 is similar to Eq. 4.15, with the only differences being the addition of $\widetilde{\boldsymbol{\sigma}}_3$ term. If $\widetilde{\boldsymbol{\sigma}}_2 = \sigma_2^0 + \boldsymbol{\sigma}_2$, using Eq. 2.11 we rewrite Eq. 4.21 as

$$\mathbf{0} = \boldsymbol{H}_t \boldsymbol{t} + \widetilde{\boldsymbol{\sigma}}_3, \text{ where} \tag{4.22}$$

$$\boldsymbol{H}_t = \begin{bmatrix} -(\boldsymbol{a}_r - \boldsymbol{\sigma}_2)^T \\ (\boldsymbol{a}_r + \boldsymbol{\sigma}_2)^\times + (a_0 - \sigma_2^0)\, \boldsymbol{I}_3 \end{bmatrix} \in \mathbb{R}^{4\times 3}. \tag{4.23}$$

Unlike the case discussed in section 4.2.1, $\boldsymbol{t}$ cannot always be directly obtained from the estimated $\widetilde{\boldsymbol{q}}_r$. This is because estimation of $\boldsymbol{t}$ would require inversion of a non-square matrix $\boldsymbol{H}_t$. As shown in section 4.3, a linear Kalman filter is employed with the following pseudo-measurement model to estimate $\boldsymbol{t}$,

$$\boldsymbol{h}_t = \boldsymbol{H}_t \boldsymbol{t} + \widetilde{\boldsymbol{\sigma}}_3, \tag{4.24}$$

and pseudo-measurement $\boldsymbol{h}_t = \mathbf{0}$, where $\boldsymbol{h}_t \in \mathbb{R}^3$. The uncertainty associated with $\boldsymbol{h}_t$ is derived in section 4.2.3.

4.2.3 Uncertainty in pseudo-measurements when using pose and position measurements

In order to estimate the uncertainties associated with the pseudo-measurements as well as the translational vector described in the previous sections, we make use of an important result from stochastic theory [146, pp. 90–91], [143, Appendix A] described in Proposition 1.

Uncertainty in pseudo-measurements when using position measurements

To find the uncertainty in the linear pseudo-measurement, we rewrite $\boldsymbol{h}$ from Eq. 4.9 in the following form

$$\begin{aligned}\boldsymbol{h} &= \boldsymbol{H}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2)\widetilde{\boldsymbol{q}}_r, \\ &= \boldsymbol{G}(\widetilde{\boldsymbol{q}}_r)\boldsymbol{v}_{\text{true}}, \quad \text{where } \boldsymbol{v}_{\text{true}} = (\boldsymbol{a}_1^T, \boldsymbol{a}_2^T, \boldsymbol{b}_1^T, \boldsymbol{b}_2^T)^T \\ &= \begin{bmatrix}\boldsymbol{G}_1 & -\boldsymbol{G}_1 & \boldsymbol{G}_2 & -\boldsymbol{G}_2\end{bmatrix} \boldsymbol{v}_{\text{true}}. \end{aligned} \tag{4.25}$$

In Eq. 4.25, $\boldsymbol{G}_1 = \begin{bmatrix} -\boldsymbol{q}_r^T \\ -\boldsymbol{q}_r^\times + q^0 \boldsymbol{I}_3 \end{bmatrix}$ and $\boldsymbol{G}_2 = \begin{bmatrix} \boldsymbol{q}_r^T \\ -\boldsymbol{q}_r^\times - q^0 \boldsymbol{I}_3 \end{bmatrix}$, where $\widetilde{\boldsymbol{q}}_r = q^0 + \boldsymbol{q}_r$ are obtained from Eq. 2.11. Eq. 4.25 is the pseudo-measurement for a noise-free sensor measurement $\boldsymbol{v}_{\text{true}}$. If $\boldsymbol{v}$ is the sensor measurement with noise $\delta\boldsymbol{v}$, then

$$\boldsymbol{v} \triangleq \boldsymbol{v}_{\text{true}} + \delta\boldsymbol{v} \tag{4.26}$$

Solving for $\boldsymbol{v}_{\text{true}}$ from Eq. 4.26 and substituting in Eq. 4.25 yields:

$$\begin{aligned}\boldsymbol{h}(\widetilde{\boldsymbol{q}}_r) &= \boldsymbol{G}(\boldsymbol{v} - \delta\boldsymbol{v}) \\ &= \boldsymbol{G}\boldsymbol{v} + \boldsymbol{\nu}_1, \end{aligned} \tag{4.27}$$

where $\boldsymbol{\nu}_1 = -\boldsymbol{G}\left(\widetilde{\boldsymbol{q}}_r\right)\delta\boldsymbol{v}$ is a zero mean noise. From Eq. 4.27, the uncertainty in the pseudo measurement $\boldsymbol{\Sigma}^{\boldsymbol{h}}$ can be obtained using Eq. 1.

The expression for $\boldsymbol{t}$ assuming perfect measurements $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ is given in Eq. 4.11. In the presence of noise in the measurements, similar to the derivation of Eq. 4.26, we obtain from Eq. 4.10

$$\begin{aligned}\boldsymbol{t} &= \frac{\boldsymbol{p}_1 + \boldsymbol{p}_2}{2} - \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\widetilde{\boldsymbol{q}}_1 + \widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right) + \boldsymbol{\nu_2}, \quad \text{where,} \\ \boldsymbol{\nu_2} &= -\frac{\delta\boldsymbol{p}_1 + \delta\boldsymbol{p}_2}{2} + \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right). \end{aligned} \tag{4.28}$$

From Eq. 4.28, $\boldsymbol{\nu_2}$ is a zero mean noise with variance $\boldsymbol{\Sigma^t} \in \mathbb{R}^{3\times3}$,

$$\boldsymbol{\Sigma^t} = \frac{\boldsymbol{\Sigma^{p_1}} + \boldsymbol{\Sigma^{p_2}}}{4} + \boldsymbol{\Sigma^\tau}, \tag{4.29}$$

where $\boldsymbol{\tau} = \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right)$, $\delta\widetilde{\boldsymbol{q}}_i = 0 + \delta\boldsymbol{q}_i$. $\boldsymbol{\Sigma^\tau}$ is computed using Eq. 1 as shown below

$$\begin{aligned}\boldsymbol{\tau} =& \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot \frac{\delta\widetilde{\boldsymbol{q}}_1 + \delta\widetilde{\boldsymbol{q}}_2}{2} \odot \widetilde{\boldsymbol{q}}_r^*\right) \\ =& \text{vec}\left(\widetilde{\boldsymbol{q}}_r \odot (0 + \boldsymbol{\sigma})\right) = -\boldsymbol{G}_2\boldsymbol{\sigma}\end{aligned}$$

where $\tilde{\boldsymbol{\sigma}} = \frac{1}{2}\begin{bmatrix}\boldsymbol{G}_3 & \boldsymbol{G}_3\end{bmatrix}\begin{bmatrix}\delta\widetilde{\boldsymbol{q}}_1 \\ \delta\widetilde{\boldsymbol{q}}_2\end{bmatrix}$ and $\boldsymbol{G}_3 = \begin{bmatrix}\boldsymbol{q}_r^\times + q^0\boldsymbol{I}_3\end{bmatrix}$ is obtained from Eq. 2.11. Eq. 1 is then used to find $\boldsymbol{\Sigma^\sigma}, \boldsymbol{\Sigma^\tau}$.

Uncertainty in pseudo-measurements when using pose measurements

For pose based measurements, there are two pseudo-measurements corresponding to estimation to $\widetilde{\boldsymbol{q}}_R$ and $\boldsymbol{t}$. Eq. 4.19 and Eq. 4.24 are rewritten in the following form

$$\boldsymbol{h}_r = \boldsymbol{G}_r\boldsymbol{u}_{\text{true}}, \tag{4.30}$$

$$\boldsymbol{h}_t = \boldsymbol{G}_t\boldsymbol{w}_{\text{true}} + \boldsymbol{\sigma}_3, \tag{4.31}$$

where $\boldsymbol{G}_r = \begin{bmatrix} q_0 & -\boldsymbol{q}_r^T & -q_0 & \boldsymbol{q}_r^T \\ \boldsymbol{q}_r & q_0\boldsymbol{I}_3 - \boldsymbol{q}_r^\times & \boldsymbol{q}_r & -q_0\boldsymbol{I}_3 - \boldsymbol{q}_r^\times \end{bmatrix}$, $\boldsymbol{u}_{\text{true}} = (a_0, \boldsymbol{a}_r^T, b_0, \boldsymbol{b}_r^T)^T$,
$\boldsymbol{G}_t = \begin{bmatrix} 0 & -\boldsymbol{t}^T & 0 & \boldsymbol{t}^T \\ \boldsymbol{t} & -\boldsymbol{t}^\times & \boldsymbol{t} & -\boldsymbol{t}^\times \end{bmatrix}$ and $\boldsymbol{w}_{\text{true}} = (a_0, \boldsymbol{a}_r^T, \sigma_2^0, \boldsymbol{\sigma}_2^T)^T$.

Eq. 4.30 and Eq. 4.31 are the pseudo-measurements for noise free sensor measurements $\boldsymbol{u}_{\text{true}}$, $\boldsymbol{w}_{\text{true}}$. If $\boldsymbol{u}$ and $\boldsymbol{w}$ are the sensor measurements with noise $\delta\boldsymbol{u}$ and $\delta\boldsymbol{w}$ respectively, then $\boldsymbol{h}_r = \boldsymbol{G}_r\boldsymbol{u} + \boldsymbol{\nu}_3$, $\boldsymbol{h}_t = \boldsymbol{G}_t\boldsymbol{w} + \boldsymbol{\sigma}_3 + \boldsymbol{\nu}_4$, where $\boldsymbol{\nu}_3 = -\boldsymbol{G}_r\delta\boldsymbol{u}$ and $\boldsymbol{\nu}_4 = -\boldsymbol{G}_t\delta\boldsymbol{w} - \delta\boldsymbol{\sigma}_3$ are zero mean noise with covariance $\boldsymbol{\Sigma}^{\boldsymbol{h}_r}$ and $\boldsymbol{\Sigma}^{\boldsymbol{h}_t}$ respectively, which can be obtained using Eq. 1.

4.3 Kalman filter equations

As shown in Eq. 4.6 and in Eq. 4.14, $\widetilde{\boldsymbol{q}}_r$ and $\boldsymbol{t}$ can be estimated in a decoupled manner. In this work, we formulate a Kalman filter that first estimates the rotation parameter $\widetilde{\boldsymbol{q}}_r$. For Case I, translation $\boldsymbol{t}$ and $\boldsymbol{\Sigma}^{\boldsymbol{t}}$ can be directly estimated from Eq. 4.11 and Eq. 4.29 upon estimating $\widetilde{\boldsymbol{q}}_r$ and $\boldsymbol{\Sigma}^{\widetilde{\boldsymbol{q}}_r}$. However for Case II, a Kalman filter is used to estimate the mean and uncertainty in translation $\boldsymbol{t}$.

The state vector of the Kalman filter that is used to estimate $\widetilde{\boldsymbol{q}}_r$ is $\boldsymbol{x}_k = \widetilde{\boldsymbol{q}}_r$, $\boldsymbol{x}_k \in \mathbb{R}^4$. The state vector is initialized with a suitable guess for mean and uncertainty. In the absence of a good initial guess, the state is initialized to $\boldsymbol{x}_0 = (1, 0, 0, 0)^T$ with a large initial uncertainty. If the uncertainty in rotation is known in terms of some other parametrizations such as Euler angles, then the uncertainty is propagated to the space of quaternions using a Jacobian mapping as shown in [155].

Since the pose to be estimated is time-invariant, the process model is static, *i.e.*, $\boldsymbol{x}_{k|k-1} = \boldsymbol{x}_{k-1|k-1}$. Upon obtaining measurements $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$, we formulate the pseudo-measurement model $\boldsymbol{h}(\boldsymbol{x}_{k|k-1}) = \boldsymbol{H}_k \boldsymbol{x}_{k|k-1}$. The observation matrix $\boldsymbol{H}_k$ is given by Eq. 4.8 for position measurements and by Eq. 4.18 for pose measurements. The measurement uncertainty $\boldsymbol{\Sigma}_k^{\boldsymbol{h}}$ is then calculated as shown in section 4.2.3 and section 4.2.3.

The state is updated using standard equations of the Kalman filter [51]

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} - \boldsymbol{K}_k \left(\boldsymbol{H}_k \boldsymbol{x}_{k|k-1} \right), \tag{4.32}$$

$$\boldsymbol{\Sigma}^{\boldsymbol{x}}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}) \boldsymbol{\Sigma}^{\boldsymbol{x}}_{k|k-1}, \quad \text{where} \tag{4.33}$$

$$\boldsymbol{K}_k = \boldsymbol{\Sigma}^{\boldsymbol{x}}_{k|k-1} \boldsymbol{H}^T \left(\boldsymbol{H} \boldsymbol{\Sigma}^{\boldsymbol{x}}_{k|k-1} \boldsymbol{H}^T + \boldsymbol{\Sigma}^{\boldsymbol{h}}_k \right)^{-1}. \tag{4.34}$$

It has already been discussed that $\widetilde{\boldsymbol{q}}_r$ is a unit quaternion; which implies that the state vector has to be a unit vector. This requirement is not enforced by the equations of the Kalman filter directly. However, there are three methods to enforce unit-normalization of state vector (1) including the constraint as an additional pseudo-measurement [154], (2) reducing the dimension of the state vector by substituting $q_0 = \sqrt{1 - q_1^2 - q_2^2 - q_3^2}$ [144], (3) normalizing the state vector at the end of each

update step [143]. The first two methods result in non-linear measurement models, which defeats our purpose of developing equations for a truly linear filter. As a result we resort to the third method of normalizing the state vector after every update and suitably scaling the uncertainty,

$$\boldsymbol{x}^*_{k|k} = \frac{\boldsymbol{x}_{k|k}}{\left\|\boldsymbol{x}_{k|k}\right\|}, \quad \Sigma^{\boldsymbol{x}^*}_{k|k} = \frac{\Sigma^{\boldsymbol{x}}_{k|k}}{\left\|\boldsymbol{x}_{k|k}\right\|^2}. \tag{4.35}$$

Such an approach has been shown to estimate efficiently in [62] and [64].

Upon estimating $\boldsymbol{x}_{k|k}$ and $\Sigma^{\boldsymbol{x}}_{k|k}$, Eq. 4.11 and Eq. 4.29 are used to estimate $\boldsymbol{t}_{k|k}$ and $\Sigma^{\boldsymbol{t}}_{k|k}$, for Case I. For Case II, we initiate another Kalman filter whose process model is static as in the case of $\widetilde{\boldsymbol{q}}_r$. The measurement model is also linear as in the case of $\widetilde{\boldsymbol{q}}_r$. The equation for the measurement model is as shown in Eq. 4.24. The observation matrix is evaluated at the estimated value of $\widetilde{\boldsymbol{q}}_r$.

4.4 Results: Sequential Estimation with Known Data Association

We apply the filtering method developed in the earlier sections to two examples: rigid registration and sensor calibration representing Case I and Case II respectively. Simulation as well as experimental results are provided in the following sections.

Rigid registration

The rigid-registration problem can be defined as finding the $SE(3)$ element that aligns points in one reference frame to the points in another reference frame. Usually points in one frame are computed from a CAD model of the object and points in the other frame are estimated from images, position sensors, laser range scanners, etc.

In this work, we use our DQF approach to register 100 points randomly sampled from the surface of a geometric model of "Stanford bunny". We first assume that the point correspondence is known and estimate the registration with DQF, whose actual

registration parameters are $\theta_x = 0, \theta_y = 0, \theta_z = 0, x = 0, y = 0, z = 0$. Note that even though dual quaternions are used to parameterize the pose, we report the results in Euler angles and Cartesian coordinates for ease of understanding.

We sample 1000 initial registration estimates uniformly drawn from large initial errors in position, for $x, y, z \in [-10000, 10000]$mm and orientation $\theta_x, \theta_y, \theta_z \in [-180, 180]$deg. From Fig. 4-1, we observe that DQF correctly estimates the registration for all the initial estimates. Following this, we perform two more experiments with noise added to the sampled points. The noise is uniformly sampled from $[-2, 2]$mm along each axis, in one case and $[-3, 3]$mm in the other. Fig. 4-1 shows that the RMS error for all the estimates is only due to the noise in the measurements and its magnitude matches well with the noise added to the points. Thus, when the point correspondences are known, DQF accurately estimates the registration parameters even in presence of very high errors in the initial estimate.

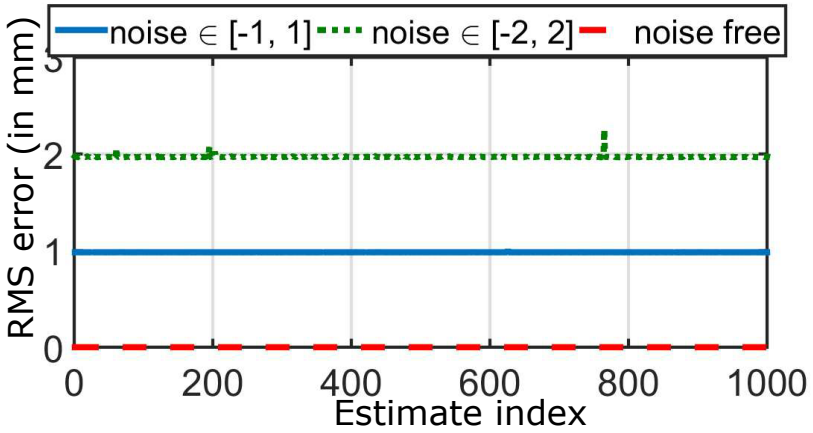


Figure 4-1: RMS error upon estimating registration parameters with DQF for 1000 runs with different initial estimates, when the point correspondence is known. Three experiments were carried out: 1) noise uniformly sampled from [-1,1], 2) noise uniformly sampled from [-2,2] and 3) no noise was added. DQF accurately estimates he registration parameters in all cases

Sensor calibration

The sensor calibration problem is as follows: given the pose of two bodies $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, defined with respect to two different inertial frames: $\{1\}$ and $\{2\}$, we would like

to estimate the rigid transformation between the two bodies, by tracking $\boldsymbol{A}_i$ and $\boldsymbol{B}_i$, where the index i denotes an instance of time. Fig. 4-2(a) shows the various frames described above. This problem can be described as $\boldsymbol{A}_{ij}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{ij}$, where $\boldsymbol{A}_{ij} = \boldsymbol{A}_i^{-1}\boldsymbol{A}_j$ and $\boldsymbol{B}_{ij} = \boldsymbol{B}_i^{-1}\boldsymbol{B}_j$. $\boldsymbol{X}$ is the rigid transformation between the two bodies which needs to be estimated as shown in Fig. 4-2(b).

If the measurements are noise-free, then $\boldsymbol{X}$ can be obtained analytically from a pair of measurements: $\boldsymbol{A}_{12}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{12}$ and $\boldsymbol{A}_{23}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{23}$ [149, 150]. But sensors are seldom noise-free, and hence several optimization based approaches exist to solve this problem [4, 156], whose solution drives many applications [157, 33, 158, 31]. Recently Faion et al [10] developed a filtering based solution to this problem, which could perform online estimation using a UKF to estimate the pose which is parameterized using axis-angle and Cartesian parameters. We compare our DQF to this UKF based estimation. We also develop an EKF based estimation using the measurement model described in [10] for a second comparison.

Simulation

We first tested our algorithm with simulated data and then with data collected from real experiments. For the simulated case, we first generate 500 random poses for the tool tip, $\boldsymbol{A}_i$ $(i = 1, \ldots, 500)$. We then choose a ground truth $SE(3)$ element $\boldsymbol{X}$ to generate the corresponding poses for the EM tracker $\boldsymbol{B}_i$. We initialize the filters to zero translation and zero rotation with an initial covariance of $\boldsymbol{\Sigma}_0^{\tilde{\boldsymbol{q}}_r} = 5\boldsymbol{I}_4$ for rotation and $\boldsymbol{\Sigma}_0^{\boldsymbol{t}} = 100\boldsymbol{I}_3$ for translation. We assume that the correspondence between the sensed poses is known. Such an assumption is reasonable as the sensor measurements can be easily time-synchronized. If this synchronization is not possible, correlation between the sensor measurements can be obtained as shown in [148].

The $SE(3)$ elements estimated by DQF, EKF and UKF are shown in Table 4.1 along with computation time for each algorithm and the error in position and orientation parameters. DQF provides faster estimates and is more accurate, especially in the translation estimation, compared to UKF and EKF.

Following this, we perturb $\boldsymbol{B}_i$ that is computed from the ground truth $\boldsymbol{X}$. The

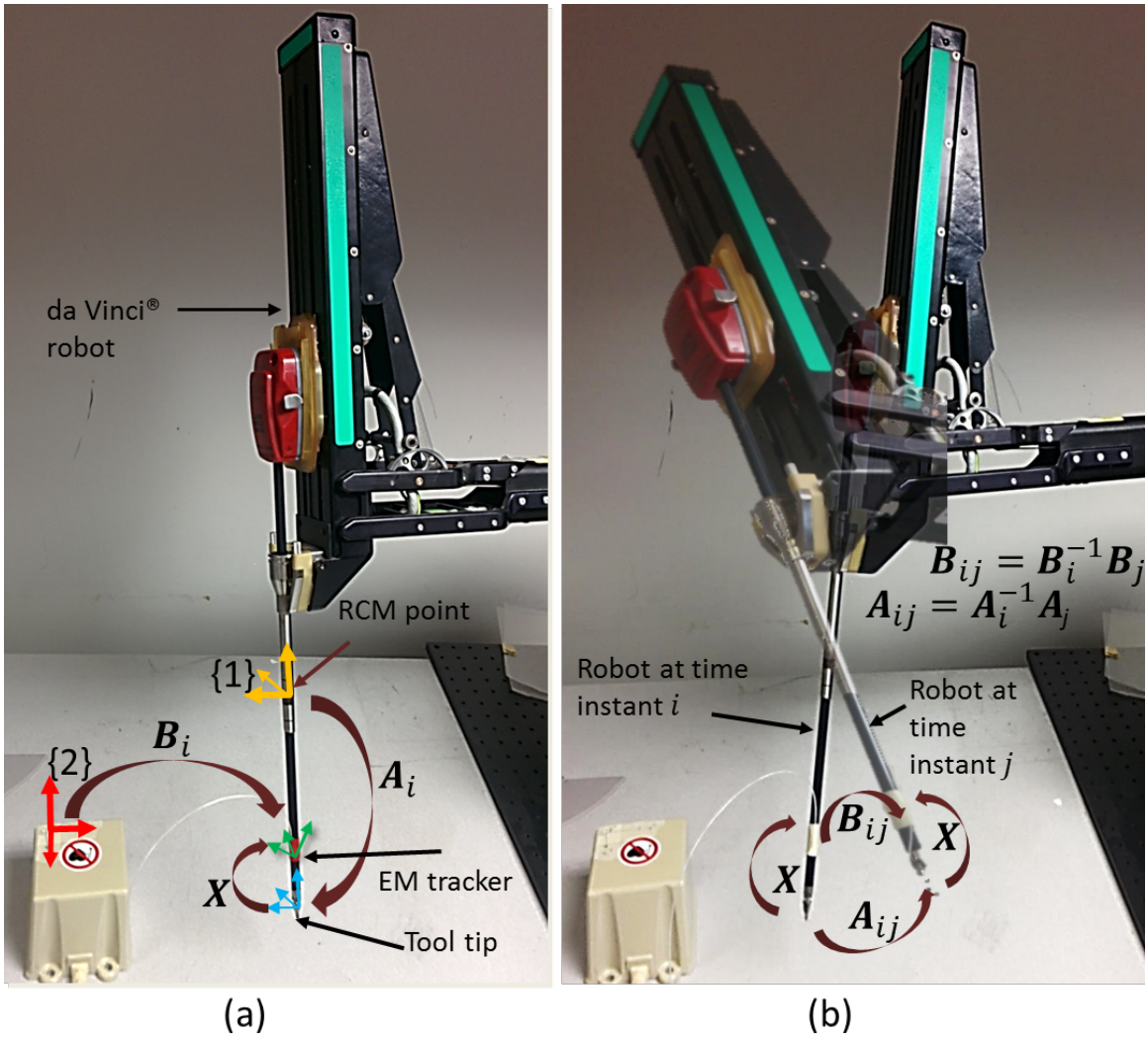


Figure 4-2: (a) The setup shows a da Vinci robot with an EM tracker rigidly attached to the tool. The reference frame for the EM tracker is shown in red. The reference frame for the robot is located at its remote center of motion (RCM), shown in yellow. The pose of the tip of the robot, $\boldsymbol{A}_i$ is shown in blue and the pose of the sensor, $\boldsymbol{B}_i$ is shown in green. $\boldsymbol{X}$ is the transformation between the tip of the robot and the EM tracker. (b) The robot is shown at two time instances i and j. $\boldsymbol{A}_{ij}$ is obtained from kinematics and $\boldsymbol{B}_{ij}$ is obtained from the EM tracker measurements. The unknown to be solved for is $\boldsymbol{X}$, which can be posed in the form: $\boldsymbol{A}_{ij}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_{ij}$.

pose is perturbed by a translation uniformly sampled from the interval $[-2\ 2]$mm along each axis and a rotation uniformly sampled from $[-10\ 10]$deg along each axis. The estimated parameters are shown in Table 4.1. Once again DQF estimated faster and is more accurate than UKF and EKF.

Experimental validation

In order to test our formulation with real data, we use an experimental setup as shown in Fig. 4-2(a), which consists of a da Vinci® surgical robot (Intuitive Surgical Inc., Mountain View, CA) and an electromagnetic (EM) tracking sensor (trakSTAR™ from Ascension Technologies, Burlington,VT). The tracker is rigidly attached to a known point on the tool of the robot. The robot is then telemanipulated and the position and orientation of the tip of the robot is measured from the kinematics. The position and orientation of the EM tracker with respect to the inertial frame attached to the magnetic field generator is also simultaneously recorded. We then use DQF, EKF and UKF to estimate the transformation between the frames of the tip of the robot and the frame of the EM tracker. The last three rows of Table 4.1 shows the parameters as estimated by DQF, EKF and UKF.

Fig. 4-3 shows the values of the quaternion and the translation vector as estimated by the dual quaternion filter. The estimated values converge at around 100 measurements for rotation and 200 measurements for translation. Since the rotation estimation does not depend upon translation estimation, we can stop running the filter that estimates rotation after convergence and continue to run the translation filter until convergence. We observe that it takes around 0.08s for the DQF estimate to converge which is roughly 5 times faster than EKF and 15 times faster than UKF, while being more accurate than both EKF and UKF. Since the estimation is close to real time, we implement this algorithm in an online manner to estimate $SE(3)$ elements as needed.

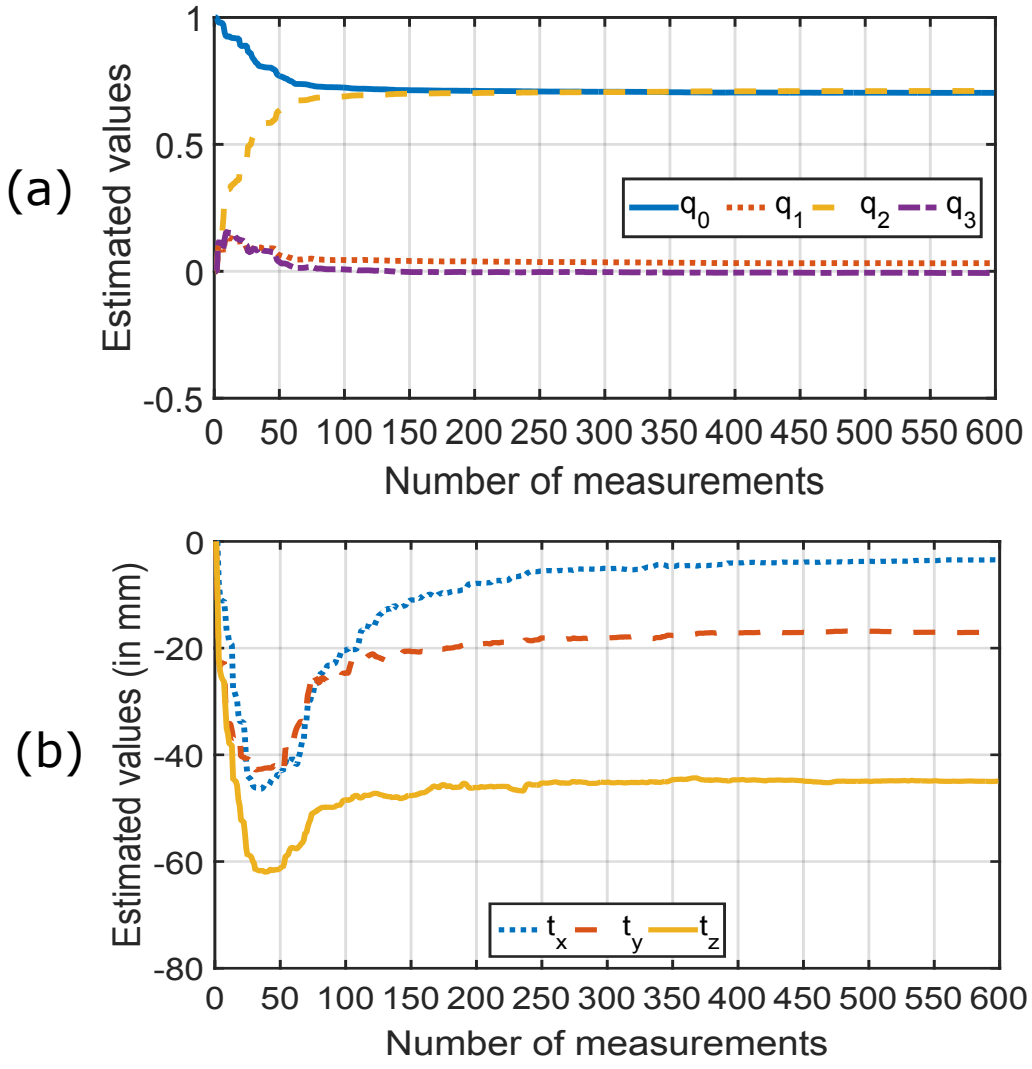


Figure 4-3: (a) The plot shows the estimated value of the quaternion that represents the rotation. The values converge at around 100 measurements. (b) The plot shows the estimated value of the translation vector. The values converge at around 200 measurements.

Table 4.1: Results for sensor calibration using dual quaternion filtering

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	Time (s)
Simulation: No noise in sensor measurements							
Actual	5.73	8.59	11.46	10.00	-16.00	35.00	–
DQF	5.73	8.59	11.46	9.96	-15.95	34.91	0.25
EKF	3.38	1.82	5.25	10.09	-15.93	35.05	1.20
UKF	3.56	10.59	10.81	9.98	-15.98	35.05	3.11
Simulation: With noise in sensor measurements							
DQF	5.59	8.22	11.38	9.95	-15.95	34.81	0.24
EKF	-3.48	4.22	8.36	10.14	-16.01	35.01	1.15
UKF	5.89	10.44	10.01	10.83	-16.81	34.81	3.20
Robot experiments							
Actual	-4	-20	45	105	88	109	–
DQF	-4.10	-17.50	-45.10	105.55	87.88	108.69	0.27
EKF	-3.60	-22.00	-45.10	105.97	86.15	107.08	1.39
UKF	-4.40	-14.1	-47.50	132.11	87.05	135.01	3.70

4.5 Results: Sequential Estimation with Unknown Data Association

We perform registration in a more realistic scenario where point correspondence is unknown. Point correspondence to the CAD model is found using a closest point rule as in [11, 6, 5]. We repeat the exercise by adding noise to the points and then estimating the transformation. The noise that is added is uniformly sampled from $[-2,\ 2]$ mm along each axis. In both the cases, we compare the dual quaternion based filtering to an EKF-based estimator [6] and a UKF-based estimator [5]. Note that these are not naive versions of EKF and UKF, but specific implementations of [6] and [5] that we use, but we shall still refer to them as EKF and UKF in this chapter.

We choose an initial guess of zero rotation and zero translation and an initial covariance of $\mathbf{\Sigma}_0^{\tilde{q}_r} = 5\boldsymbol{I}_4$ for rotation and $\mathbf{\Sigma}_0^{\boldsymbol{t}} = 100\boldsymbol{I}_3$ for translation. DQF, EKF and UKF are implemented with 40 initial starts obtained by locally perturbing the initial guess by translation sampled uniformly from $[-15,\ 15]$mm along each axis and a rotation sampled uniformly from $[-30,\ 30]^\circ$ along each axis. Since the problem has

several local minima, using multiple initial guesses improves the chances of finding the global minimum.

Fig. 4-4(a) shows the CAD model of the Stanford bunny in green. The blue diamond markers show the initial guess for the location of the points and the red circular markers show the DQF estimated location of the points. Fig. 4-4(b) shows the RMS error versus number of points used to estimate the parameters. The RMS error decreases with the usage of more point measurements. DQF and EKF converge to a smaller RMS error at about 10 points, while the UKF takes many more points to converge. The first four rows of Table 4.2 show the actual registration parameters and the estimated registration parameters. The right column of Table 4.2 shows the time taken by the filters to update for 100 point measurements. DQF converges an order of magnitude faster than EKF and UKF and also has the lowest RMS error.

Table 4.2: Results for registration using dual quaternion filter

No noise	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)	Time (s)
Actual	22	-23	20	15	-10	-10	–	–
DQF	22.54	-21.52	20.03	17.28	-9.94	-10.15	1.12	1.17
EKF	22.35	-26.39	21.11	11.43	-11.44	-14.76	3.88	155.02
UKF	21.36	-23.89	18.94	16.39	-5.95	-10.55	2.47	247.56
With noise								
DQF	22.34	-24.22	18.79	13.37	-9.09	-10.18	2.70	1.08
EKF	20.29	-26.09	20.69	8.76	-12.79	-8.08	3.81	324.23
UKF	20.08	-24.78	14.6	11.90	-6.08	-8.04	4.80	510.73

Fig. 4-5 shows the results for the case where the sampled points are corrupted with a noise uniformly sampled from $[-2\ 2]$mm along each axis. DQF accurately registers the points to the CAD model as shown in the last three rows of Table 4.2. DQF once again performs better than EKF and UKF, and takes lower computational time [1].

We do not deal with outliers in this chapter. However, we report the results for various levels of outliers in Table 4.3. We repeat the experiment from the noisy

[1]The computational time taken is calculated for a code running on MATLAB R2015a software from MathWorks, running on a ThinkPad T450s computer with 8 GB RAM.

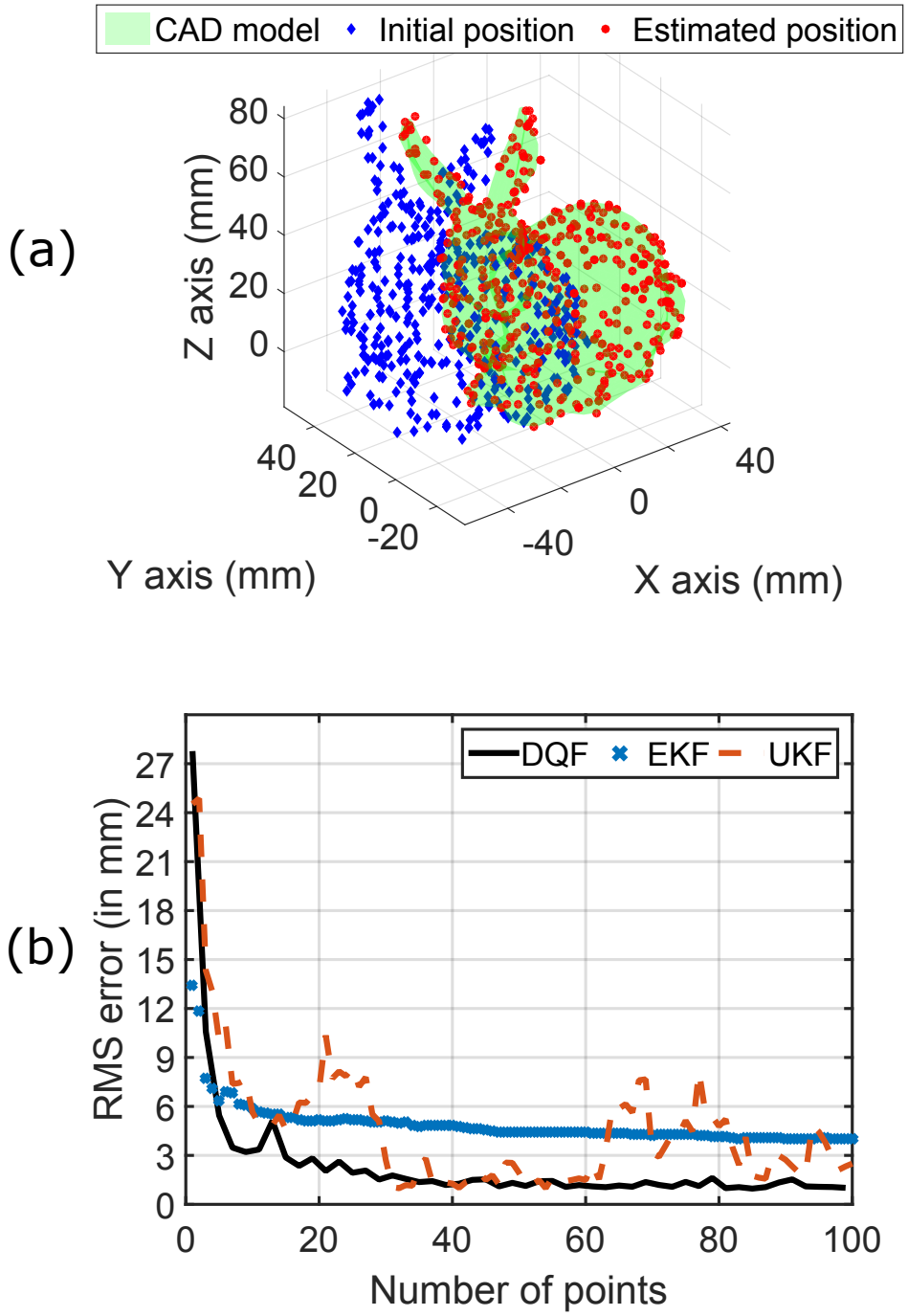


Figure 4-4: (a) Initial position and DQF estimated position of 100 points are shown against the CAD model of the "Stanford bunny". DQF accurately registers the points to the CAD model. (b) A plot of the RMS error wrt number of points for DQF, EKF and UKF. DQF and EKF converge quickly, while UKF takes a while to converge. DQF however converges to lower RMS error, with computation time an order of magnitude lower than EKF and UKF as shown in Table 4.2.

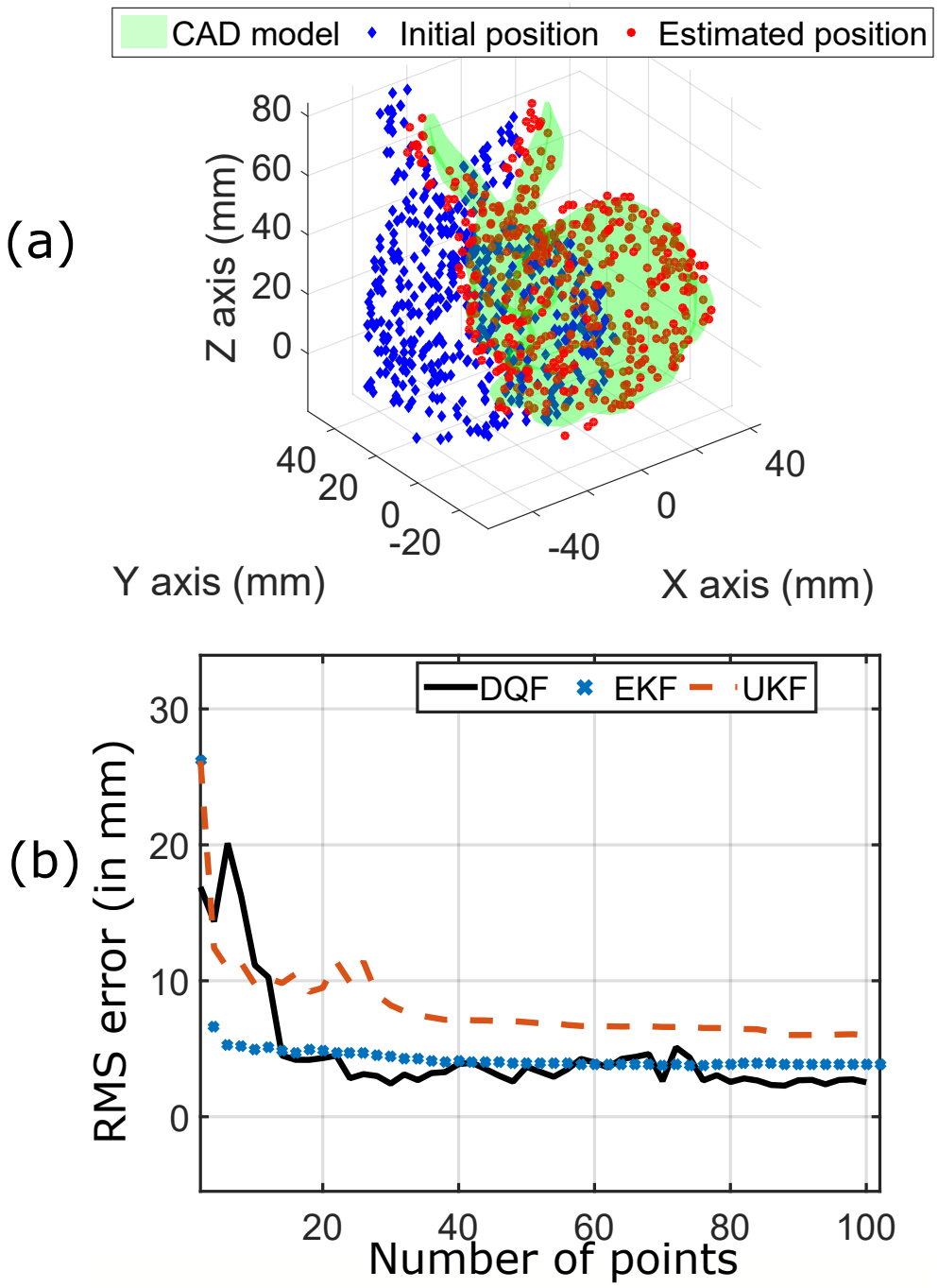


Figure 4-5: (a) Initial position and estimated position of 100 points with added noise are shown against the CAD model of the "Stanford bunny". DQF estimates the registration parameters accurately even in the presence of noise. (b) A plot of the RMS error wrt number of points for DQF, EKF and UKF. DQF and EKF converge quickly, but UKF takes a while to converge. Overall, all the three filters converge closely to one another, with the DQF performing marginally better. The DQF converges with computation time an order of magnitude lower than the other two as shown in Table 4.2.

measurement case of Table 4.2 by adding various percentage of outliers. From the available measurements, we randomly pick a chosen percentage of measurements and add a noise which is randomly obtained from $[-100,\ 100]$mm along each axis, to simulate outliers. As expected with increase in the level of outliers, the performance of the algorithm degrades. We provide a discussion on dealing with outliers in Chapter 6.

Table 4.3: Registration with DQF in the presence of outliers

Outlier %	RMS (mm)
0	2.70
5	4.74
10	8.45
15	11.56
20	15.87

4.6 Conclusion and Discussion

In this chapter, we have developed linear measurement models to be used with Kalman filters for pose estimation. This was possible due to our choice of using dual quaternions to represent $SE(3)$ elements and combining multiple sensor measurements simultaneously. All the information contained in the non-linear update model was encoded in the linear measurement model and its corresponding uncertainty, which happens to be state dependent in this case. Since the dependence on the state was found to be linear, results from stochastic theory were used to determine the exact expressions for the uncertainty. We show that the new linear measurement model allows for decoupled estimation of rotation and translation using independent Kalman filters. The decoupled estimation potentially has the advantage of running in-parallel and accelerating the estimation process.

We have shown through simulations and experiments that the dual quaternion-based linear filtering (DQF) is capable of estimating the pose more accurately with less computation time compared to state-of-the-art filtering methods for pose estimation. These characteristics of the DQF, make it an ideal candidate to be used in applications

that require real-time pose estimation such as sensor calibration, localization and manipulation. We do not perform any systematic study for dealing with outliers in this chapter. We present ways to deal with outliers in Chapter 6.

A major limitation of the DQF is that, only the mean of the updated state is guaranteed to lie on a unit-hypersphere. If we were to sample a point from the PDF of the updated state, it would not satisfy the unit-norm constraint required for a rotation quaternion. Furthermore, the fact that $\widetilde{\boldsymbol{q}}_r$ and $-\widetilde{\boldsymbol{q}}_r$ produce the same rotation is not captured by the unimodal Gaussian distribution used to model the state. These limitations will be addressed later in Chapter 6.

4.7 Contribution

The contributions from this chapter include:

1. Development of a linear update model for pose estimation. The linear model is probabilistic and we derive exact expressions for the mean and uncertainty of this model. The linear update model has be derived for position and pose measurements.

2. Development of a Kalman filter that uses the linear update model for fast and robust pose estimation.

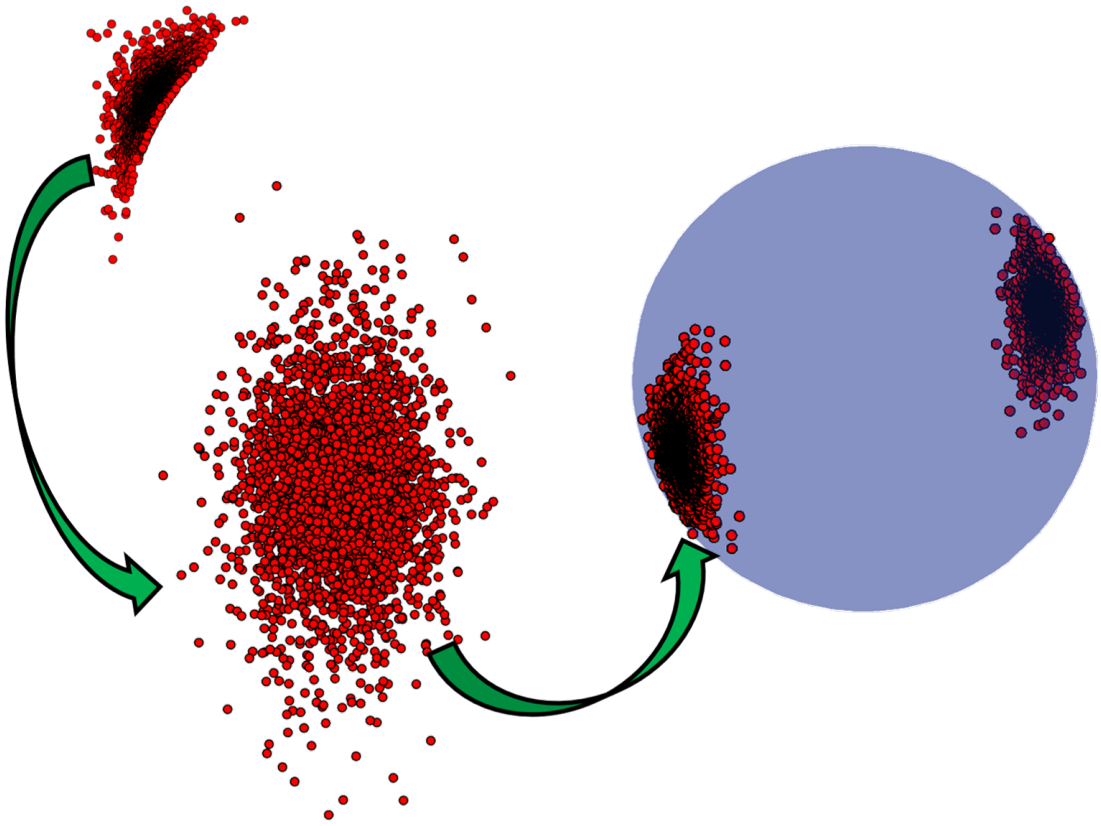
4.8 Published Work

Material from this chapter has appeared in the following publication

1. R Arun Srivatsan, Gillian T. Rosen, Feroze M. Naina, and Howie Choset,"Estimating SE(3) elements using a dual-quaternion based linear Kalman filter, in the proceedings of Robotics: Science and Systems, Michigan, USA, June 2016, doi 10.15607/RSS.2016.XII.013.

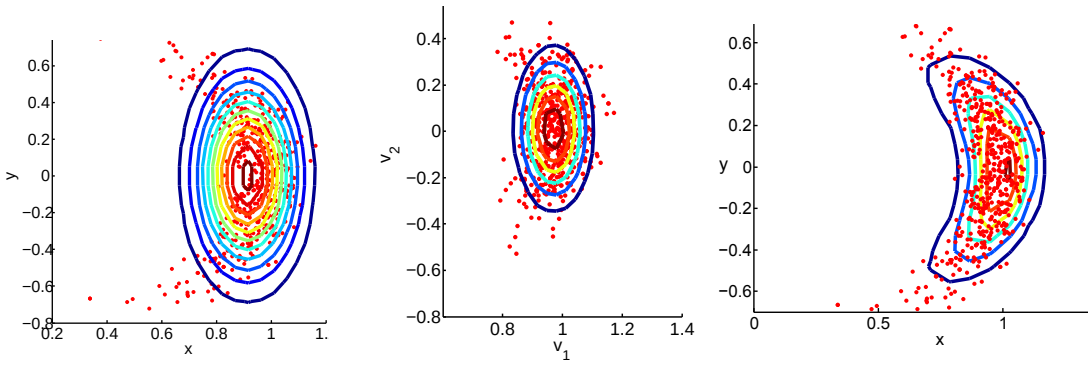
Part II

Finding the right parameter space for estimation



Chapter 5

Lie Algebra Filter for Pose Estimation



(a) (b) (c)

Figure 5-1: The red points show the final location of a differentially driven robot moving along a straight line, with noisy wheel speeds. The PDF contours of Gaussian of best fit (a) in Cartesian coordinates (b) in exponential coordinates (c) in exponential coordinates when mapped into Cartesian space.

The $SE(3)$ element can be parameterized in several ways, for example using Cartesian coordinates and Euler angles, dual quaternions, or using exponential coordinates. However, using the exponential coordinates provides a 'natural' way of describing pose uncertainties using Gaussians as they inherently exist in a Euclidean space [159]. For example, Fig 5-1(a) shows the distribution over the Cartesian coordinates for a differentially driven robot moving along a straight line. Note how the PDF contours of

Gaussian of best fit are not a good approximate to the distribution. On the other hand, representing the state vector in terms of exponential coordinates, captures the distribution better. Fig 5-1(b) shows how the PDF contours of Gaussian of best fit in exponential coordinates, approximates the distribution closely. When mapped back into the Cartesian space through the exponential map, the Gaussian contours appear to bend and take a banana-like shape as shown in Fig. 5-1(c), and approximate the distribution closely.

Exponential coordinates have been extensively used to parameterize $SE(3)$ in a number of robotics problems [88, 160, 161, 1]. A number of recent works perform ltering on Lie groups using exponential coordinates [162, 163, 164], [165, 166]. Taking inspiration from the literature, in this chapter we show that deriving measurement models for pose estimation using Gaussian distributions in exponential coordinates provides accurate pose estimation even in the nonlinear setting. We exploit the underlying structure of the Lie group and algebra to estimate the pose and shape of a medical snake robot (shown in Fig. 5-2) as well as perform SCAR using SCAR-LSQ-CMU which was described in Chapter 3.

5.1 Pose and shape estimation for a medical snake robot

When compared to conventional procedures, MIS have potential benefits including "reduced-pain", minimal "blood-loss", faster healing and reduced tissue disruption. To further improve upon MIS, our group, along with others [167, 168, 169, 170] have been developing snake-like robots to provide deeper access to the anatomy with fewer incisions. However, robot-guided MIS introduces other drawbacks such as lack of situational awareness and limited sensory feedback. This forces surgeons to use their expertise and cognitive reasoning in finding relation between pre-operative images and the surgical reality. Although emerging technologies such as fiber optics, MRI, ultrasound (US) and CT have proved to be useful, they suffer from a number of

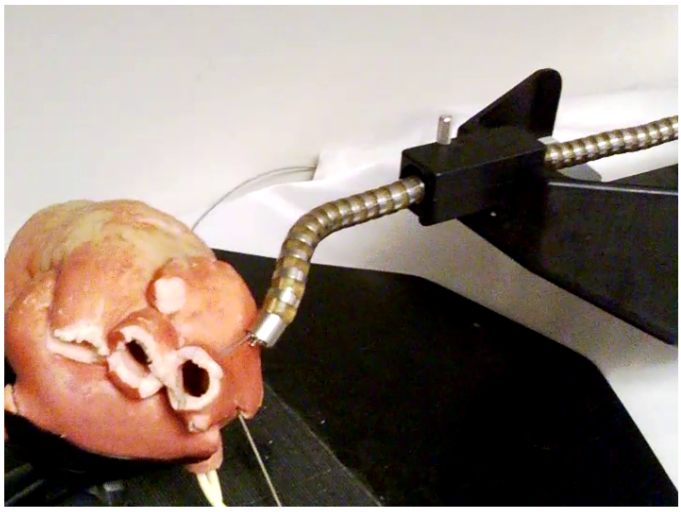


Figure 5-2: A highly articulated robotic probe, whose shape and pose is estimated in this chapter

drawbacks. MRI restricts the robot from carrying any magnetic material and often requires redesigning robots to be MRI compatible [171]. Using CT scan for localizing the robot can result in over exposure to harmful radiation. US images are very noisy and often difficult to work with. Fiber optics have been used not only to provide a first person view form the tip of the robot, but also to find the shape of the robot by using fiber Bragg grating (FBG) [172]. The images from fiber optics have a limited field of view and the shape sensing using FBG is very expensive and is also affected by changes in temperature as observed by Mihailov [173].

Prior efforts include generating 3D models of anatomical structures from pre-operative scans and combining them with model-based tool tracking, to create a virtual view of the operation for visual feedback. To successfully use this technique, the tool needs to be registered in the frame of reference of the 3D model. A number of algorithms have been developed to take input from electro-magnetic sensors [174], ultrasound [175, 176] etc., and help in the registration process.

When the tool is a highly articulated robot, estimation is more difficult because of the additional degrees of freedom (DOFs). Also complete knowledge of its shape configuration is necessary to avoid interfering with organs. In the work of Tully *et. al.* [3]

a nonlinear stochastic filtering method is used to estimate the shape and configuration of a snake robot (see Fig. 5-2). It measures the pose of the tip and prescribes a motion model based on the robot's kinematics and history of the previous inputs. The state of the snake robot, both its base pose and its internal shape, is described in terms of Cartesian coordinates and Euler angles. In presence of high uncertainties, this parametrization does a poor job of estimating the shape of the robot. However, recent work by Long *et. al.* [1] shows that modeling a system using exponential coordinates yields better estimation as opposed to using conventional parameters especially in the presence of large uncertainties.

Exploiting the underlying structure of the Lie group and algebra which defines motions for the robot, in this work we use an extended Kalman filter to estimate the state of the system that is described by exponential coordinates. This approach provides better estimate of shape than using a conventional parametrization [177].

5.1.1 Problem Formulation

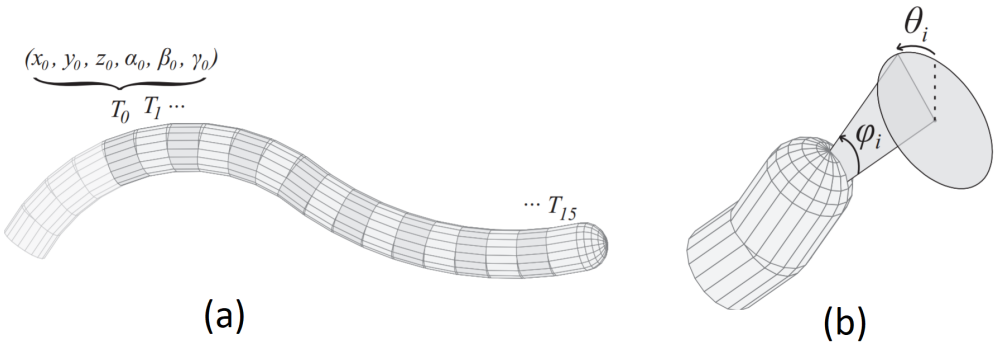


Figure 5-3: (a) The state parameterization used to define the configuration of the snake robot. (b) The effect of angles θ_i and ϕ_i on the pose between two successive links. The images are taken from [3].

Definition of state vector

The state vector for an N link snake robot (see Fig. 5-3(a)) is described by Tully *et. al.* [3] as

$[\boldsymbol{p}_0, \boldsymbol{e}_0, \phi_1, \theta_1, \ldots, \phi_{N-1}, \theta_{N-1}]^T$, where $\boldsymbol{p}_0 = [x_0, y_0, z_0]^T$ and $\boldsymbol{e}_0 = [\alpha_0, \beta_0, \gamma_0]^T$ define

the position and orientation respectively of the most proximal link, while ϕ_i, θ_i give the rotation between successive links (see Fig. 5-3(b)). In this work, the state vector is formulated in terms of exponential coordinates as,

$$\boldsymbol{x}_k = [\underbrace{v_1^0, v_2^0, v_3^0}_{\boldsymbol{v}}, \underbrace{\omega_1^0, \omega_2^0, \omega_3^0}_{\boldsymbol{\omega_0}}, \omega_2^1, \omega_3^1, \ldots, \omega_2^{N-1}, \omega_3^{N-1}]^T, \tag{5.1}$$

where $(\boldsymbol{v}^T, \boldsymbol{\omega_0}^T)^T$ is the twist vector that is used to identify the element $\boldsymbol{T}_0 \in SE(3)$ of the most proximal link of the robot. The variables ω_2^i, ω_3^i for each link i are exponential coordinates, representative of the relative rotation between successive links with the subscripts indicating the axis along which they are defined (see Fig. 5-3(b)). This section establishes the relation between the state vector in [3] and the one in Eq. (5.1). In doing so we define the notion of state for both the proximal and the remaining links.

Proximal link

Given the state vector of the proximal link, $[x_0, y_0, z_0, \alpha_0, \beta_0, \gamma_0]^T$, the rotation matrix is first obtained as $\boldsymbol{R} = \boldsymbol{R_z}(\alpha_0)\boldsymbol{R_y}(\beta_0)\boldsymbol{R_x}(\gamma_0)$, where $\boldsymbol{R_k}(\phi)$ is the rotation matrix describing the rotation about axis $\boldsymbol{k}$ by an angle ϕ. Upon obtaining $\boldsymbol{R}$, the matrix logarithm of $\boldsymbol{R}$ is evaluated to obtain the rotational component of exponential coordinates $\boldsymbol{\omega^0} = [\omega_1^0, \omega_2^0, \omega_3^0]^T$ as shown in [88]. We obtain the following:

$$\widehat{\boldsymbol{\omega^0}} = \frac{\theta}{2\sin\theta}(\boldsymbol{R} - \boldsymbol{R}^T), \tag{5.2}$$

where $\theta = \arccos\left(\frac{\text{trace}(\boldsymbol{R})-1}{2}\right)$ and $\boldsymbol{R} \neq \boldsymbol{I}_{3\times3}$. Note that, if $\boldsymbol{R} = \boldsymbol{I}_{3\times3}$, then $\theta = 2\pi k$, where k is any integer and $\boldsymbol{\omega^0}$ can be chosen arbitrarily. Having obtained $\boldsymbol{\omega}^0$, $\boldsymbol{v} = [v_1, v_2, v_3]^T$ can obtained as follows:

$$\boldsymbol{v} = \boldsymbol{W}[x_0, y_0, z_0]^T, \tag{5.3}$$

where $\boldsymbol{W} = \boldsymbol{I}_{3\times3} - \frac{\widehat{\boldsymbol{\omega}\mathbf{0}}}{2} + \left(1 - \frac{\theta \sin\theta}{2(1-\cos\theta)}\right) \frac{\widehat{\boldsymbol{\omega}\mathbf{0}}^2}{\theta^2}$. Note that if $\theta = 0$, then $\boldsymbol{W} = \boldsymbol{I}_{3\times3}$.

Distal links

The relative rotation between successive links is represented by ϕ_i and θ_i in Fig. 5-3(b). Let the exponential coordinates describing the relative motion between successive links be $[v_1^i, v_2^i, v_3^i, \omega_1^i, \omega_2^i, \omega_3^i]^T$. The equivalent of this in the exponential coordinates can be obtained by following the procedure used in the case of the proximal link. The relative motion between links is purely rotational and so we have, $v_1^i = 0, v_2^i = 0, v_3^i = 0$. Further since there is no roll motion between successive links, $\omega_1^i = 0$. Thus ω_2^i, ω_3^i alone are sufficient to describe the relative motion. Taking advantage of the way ϕ_i, θ_i are defined, it is equivalent to stating that, in the axis-angle representation, i^{th} link is oriented at an angle ϕ_i about an axis: $[0, \cos\theta_i, \sin\theta_i]^T$, with respect to $(i-1)^{\text{th}}$ link.

It is well know that given an axis $\boldsymbol{k} = [k_x, k_y, k_z]^T$ and an angle of rotation ϕ about the axis, $\boldsymbol{R} = \exp(\boldsymbol{K}\phi)$, where $\exp(\cdot)$ is the matrix exponential, $\boldsymbol{K}$ is the skew-symmetric matrix formed from the vector $\boldsymbol{k}$ and the corresponding exponential coordinates are $[k_x\phi, k_y\phi, k_z\phi]^T$ [87]. Hence the required exponential coordinates describing the relative motion between successive links can be obtained in terms of ϕ_i, θ_i as follows:

$$\omega_2^i = \phi_i \cos\theta_i, \tag{5.4}$$

$$\omega_3^i = \phi_i \sin\theta_i. \tag{5.5}$$

Thus we establish that for an $N-$link snake, the state vector used in this work is of the same dimensionality as that used in the work of Tully *et. al.* [3] and the two representations are equivalent.

Obtaining the $SE(3)$ element

Given the state vector $\boldsymbol{x}_k$, the transformation matrix describing the pose of the most proximally located link is given by:

$$\boldsymbol{T}_0(\boldsymbol{x}_k) = \begin{pmatrix} \boldsymbol{R}_0 & \boldsymbol{p}_0 \\ \boldsymbol{0}_{1\times 3} & 1 \end{pmatrix}, \tag{5.6}$$

where,

$$\begin{aligned} \boldsymbol{R}_0 &= \boldsymbol{I}_{3\times 3} + \frac{\sin\theta}{\theta}\widehat{\boldsymbol{\omega}^{\boldsymbol{0}}} + \frac{1-\cos\theta}{\theta^2}\widehat{\boldsymbol{\omega}^{\boldsymbol{0}}}^2, \\ \boldsymbol{p}_0 &= \boldsymbol{V}[v_1^0, v_2^0, v_3^0]^T, \\ \boldsymbol{V} &= \boldsymbol{I}_{3\times 3} + \frac{1-\cos\theta}{\theta^2}\widehat{\boldsymbol{\omega}^{\boldsymbol{0}}} + \frac{\theta-\sin\theta}{\theta^3}\widehat{\boldsymbol{\omega}^{\boldsymbol{0}}}^2, \\ \theta &= \text{norm}(\omega_1^0, \omega_2^0, \omega_3^0). \end{aligned} \tag{5.7}$$

To compute the transformation matrix $\boldsymbol{T}_i(\boldsymbol{x}_k)$ that represents the pose of i^{th} link, the following recursive process is defined:

$$\boldsymbol{T}_i(\boldsymbol{x}_k) = \boldsymbol{T}_{i-1}\boldsymbol{T}_{i,ang}(\boldsymbol{x}_k)\boldsymbol{T}_{adv}, \tag{5.8}$$

$$\boldsymbol{T}_{i,ang}(\boldsymbol{x}_k) = \begin{pmatrix} \boldsymbol{R}_i & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{pmatrix}$$

$$\boldsymbol{T}_{adv} = \begin{pmatrix} 1 & 0 & 0 & L \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where L is the length of a link. The Cayley-Klein parameters c_2, c_3 (see Section 2.3 of Shoham *et. al.* [59]) are related to ω_2^i, ω_3^i as follows:

$$c_2 = \cos\theta \tan\frac{\phi}{2}, \tag{5.9}$$

$$c_3 = \sin\theta \tan\frac{\phi}{2}, \tag{5.10}$$

where, $\theta = \text{atan2}(\omega_3^i, \omega_2^i)$, $\phi = \sqrt{(\omega_2^i)^2 + (\omega_3^i)^2}$. The rotation matrix $\boldsymbol{R}_i$ can be obtained from c_2, c_3 as follows:

$$\boldsymbol{R}_i = \begin{pmatrix} 1 - c_2^2 - c_3^2 & -2c_3 & 2c_2 \\ 2c_3 & 1 + c_2^2 - c_3^2 & 2c_2c_3 \\ -2c_2 & 2c_2c_3 & 1 - c_2^2 + c_3^2 \end{pmatrix}. \tag{5.11}$$

5.1.2 Motion and measurement model

The motion of the snake-robot has three distinct modes: advancing, retracting and steering. This section describes the motion models, $f_a(\boldsymbol{x}_k)$, $f_r(\boldsymbol{x}_k)$, $f_s(\boldsymbol{x}_k)$ respectively for the same. An electro-magnetic sensor is used to measure the position of the tip of the robot. A forward kinematic measurement model that incorporates measurement from the sensor is also described in this section.

Advancing and Retracting motion

When the snake is advanced by one link, the state space grows by two parameters due to addition of $\omega_2^{N-1}, \omega_3^{N-1}$. During advancement there is no relative motion between the most distal link and the link before it, therefore $\phi_{N-1} = \theta_{N-1} = 0$, which implies $\omega_2^{N-1} = \omega_3^{N-1} = 0$. Thus the motion model for advancement, $f_a(\boldsymbol{x}_k)$ remains the same as the one defined by Tully *et. al.* [3]:

$$f_a(\boldsymbol{x}_k) = [\boldsymbol{x}_k^T, 0, 0]^T. \tag{5.12}$$

Assuming M is the length of the state vector at time-step k, the motion model for retraction is also similar to the one defined by Tully *et. al.* [3]:

$$f_r(\boldsymbol{x}_k) = [\boldsymbol{I}_{(M-2\times M-2)}\boldsymbol{0}_{(M-2\times 2)}]\boldsymbol{x}_k. \tag{5.13}$$

The length of the state is reduced by two because the EKF would track one link less.

Steering motion

Tully *et. al.* [3] provide the relation between the differential lengths of the cables (d_1, d_2, d_3) and the orientation of the most distal link relative to the link behind it. Since there exists a typographical error in the equations listed there and a derivation of the relation is not provided, the correct relation has been derived and included here for the sake of completeness.

Steering model

The kinematic representation of two adjacent units of the snake robot is shown in Fig. 5-4. The units can be approximated by equilateral triangles of circumradius r whose centers are separated by a fixed distance l_c. The three cables pass through the three vertices of the equilateral triangles at the base and the top. Kinematically these can be modeled as prismatic actuators. The coordinates of the vertices of the equilateral triangle of the base link are given as:

$$\boldsymbol{b}_1 = [0, 0, r]^T, \tag{5.14}$$

$$\boldsymbol{b}_2 = \left[0, -\frac{\sqrt{3}r}{2}, -\frac{r}{2}\right]^T, \tag{5.15}$$

$$\boldsymbol{b}_3 = \left[0, \frac{\sqrt{3}r}{2}, -\frac{r}{2}\right]^T. \tag{5.16}$$

The coordinates of the vertices of the equilateral triangle of the top link are, given a relative rotation of $\boldsymbol{R}_i$ as defined in Eq. (5.11):

$$\boldsymbol{p}_k = \boldsymbol{R}_i(\boldsymbol{b}_k + [l_c, 0, 0]^T), \quad \text{for } k = 1,\, 2,\, 3. \tag{5.17}$$

The length of the cables $d_i,\ i = 1, 2, 3$ can be obtained geometrically as the distance between $\boldsymbol{b}_i$ and $\boldsymbol{p}_i$:

$$\eta_k \triangleq (\boldsymbol{b}_k - \boldsymbol{p}_k)^T(\boldsymbol{b}_k - \boldsymbol{p}_k) - d_k^2 = 0, \tag{5.18}$$

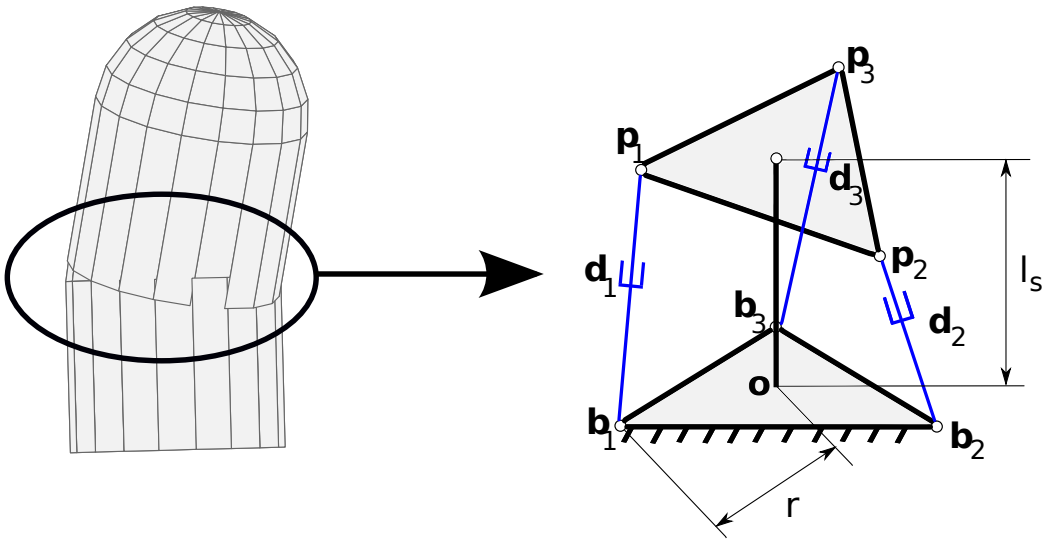


Figure 5-4: Kinematic representation of two adjacent links

where $k = 1, 2, 3$. The design of the robot is such that, the ratio $\frac{l_c}{r} \approx 0$. Hence, substituting this condition and simplifying yields the following constrain equations:

$$\eta_1 \triangleq 4r^2(\cos\theta_{N-1})^2 \left(\sin\frac{\phi_{N-1}}{2}\right)^2 - d_1^2 = 0 \tag{5.19}$$

$$\eta_2 \triangleq -r^2(-2 + \cos 2\theta_{N-1} + \sqrt{3}\sin 2\theta_{N-1}) \left(\sin\frac{\phi_{N-1}}{2}\right)^2 - d_2^2 = 0 \tag{5.20}$$

$$\eta_3 \triangleq r^2(2 - \cos 2\theta_{N-1} + \sqrt{3}\sin 2\theta_{N-1}) \left(\sin\frac{\phi_{N-1}}{2}\right)^2 - d_3^2 = 0. \tag{5.21}$$

Since there are only two unknowns, θ_{N-1}, ϕ_{N-1} and three equations, any two equations can be solved simultaneously to obtain expressions for θ_{N-1}, ϕ_{N-1}. Eliminating $\sin(\phi_{N-1}/2)^2$ from Eq. (5.19) and Eq. (5.20), one obtains:

$$\eta_4 \triangleq -2d_1^2 + 2d_2^2 + (d_1^2 + 2d_2^2)\cos 2\theta_{N-1} + \sqrt{3}d_1^2 \sin 2\theta_{N-1} = 0. \tag{5.22}$$

Eq. (5.22) is in terms of $\sin 2\theta_{N-1}$ and $\cos 2\theta_{N-1}$, by using half-tangent substitution, it can be converted to an equation wholly in terms of $\tan\theta_{N-1}$. Upon solving for

$\tan\theta_{N-1}$, one obtains:

$$\tan\theta_{N-1} = \frac{\sqrt{3}d_1 + 2\sqrt{3}d_2}{3d_1},$$
$$\Rightarrow \theta_{N-1} = \arctan\left(\frac{\sqrt{3}d_1 + 2\sqrt{3}d_2}{3d_1}\right). \tag{5.23}$$

From Eq. (5.19), we obtain:

$$\phi_{N-1} = 2\left|\arcsin\left(\frac{d_1}{2r\cos\theta_{N-1}}\right)\right|. \tag{5.24}$$

Using Eq. (5.23) and Eq. (5.24), the change in angles from the previous time step, $\Delta\theta_{N-1}, \Delta\phi_{N-1}$ are computed and the corresponding $\Delta\omega_2^{N-1}, \Delta\omega_3^{N-1}$ are obtained using Eq. (5.4). Thus the motion model for steering the snake is:

$$f_s(\boldsymbol{x}_k) = \boldsymbol{x}_k + [\mathbf{0}^T_{M-1\times 1}, \Delta\omega_2^{N-1}, \Delta\omega_3^{N-1}]^T, \tag{5.25}$$

assuming M is the length of the state vector at that instant.

Measurement model

The tip of the highly articulated robot described by Degani *et. al.* [178] is sensed by inserting an electromagnetic position sensor into one of the tool channels of the robot. The sensor used in this work is a trakSTAR™ (Ascension Technologies, Burlington, VT, USA) which has the capability to measure 6-DOFs pose of the tip of the robot with respect to a world frame. Since the tracker might be removed from time to time for insertion of tools into the same channel, the roll parameter sensed, is set free. The sensor therefore observes five elements of the pose of the distal link, and the measurement model is given as:

$$h(\boldsymbol{x}_k) = [\boldsymbol{p}^T_{N-1}, \alpha_{N-1}, \beta_{N-1}]^T, \tag{5.26}$$

where $\boldsymbol{p}_{N-1}$ is the position of the distal link described in terms of Cartesian coordinates and $\alpha_{N-1}, \beta_{N-1}$ are the yaw and pitch of of the distal link as measured by the sensor. The parameters $\boldsymbol{p}_{N-1}^T, \alpha_{N-1}, \beta_{N-1}$, can be obtained from $\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)$:

$$\begin{aligned}
\boldsymbol{p}_{N-1} &= [\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{1,4}, \boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{2,4}, \boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{3,4}]^T, \\
\alpha_{N-1} &= \text{atan2}(\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{(2,1)}, \boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{(1,1)}), \\
\beta_{N-1} &= \text{atan2}(-\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{(3,1)}, \sigma),
\end{aligned}$$

where $\sigma = \sqrt{\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{(3,2)}^2 + \boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{(3,3)}^2}$ and $\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)_{i,j}$ refers to the ij^{th} term of $\boldsymbol{T}_{N-1}(\boldsymbol{x}_k)$.

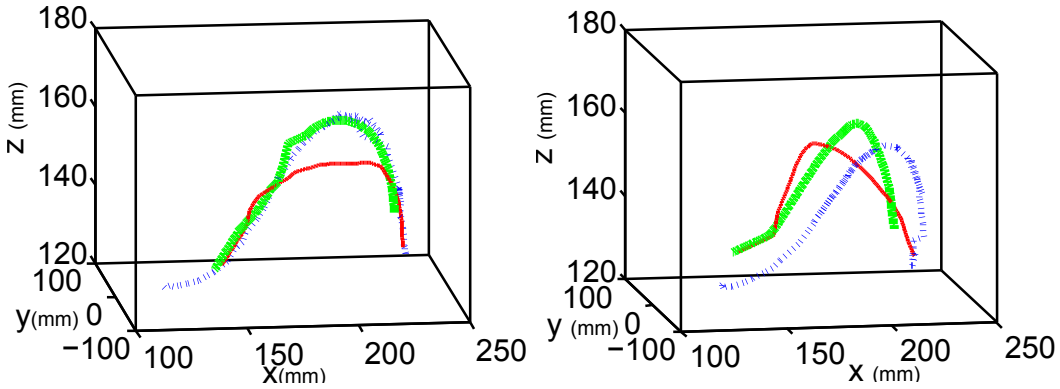
In this work the method to estimate the state of the snake given the measurements of the position and orientation of the distal tip, is similar to the one described by Tully *et. al.* [3]. The only difference is that the elements of the state vector are now in terms of exponential coordinates. It is also worth noting that we still require an advancement of at least one link before steering it for the system to be fully observable.

5.1.3 Results for robot pose and shape estimation

Two bench-top tests were conducted to find out the effectiveness of this filtering approach. The snake robot as shown in Fig. 5-2, was driven using a joystick and the distal link was tracked using an electro-magnetic sensor. During the experiment time-stamped input values from the joystick as well as from the sensor were noted. After the end of the experiment, keeping the shape of the snake fixed, the sensor was pulled out of the tool-channel and a trail of data points was recorded that could be post processed and used as ground-truth.

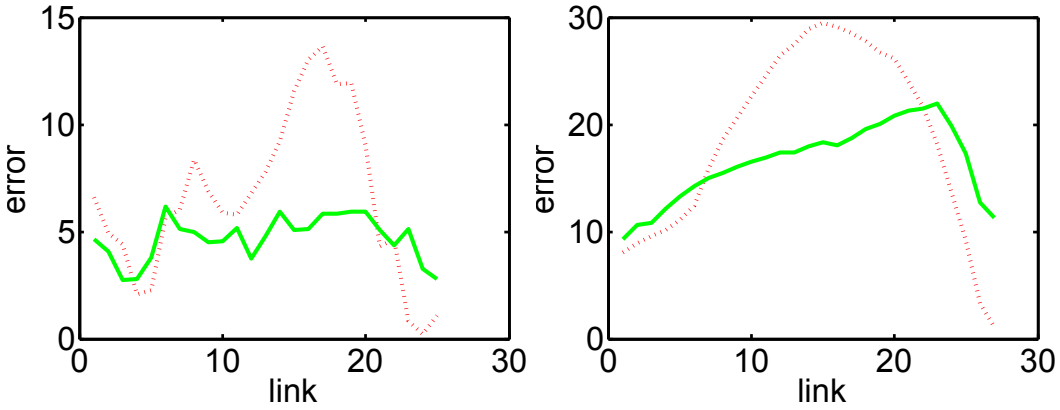
Using the filtering approach discussed in Section. 5.1.2, the shape of the robot was estimated in both the tests. The length of each link was measured to to be 6.9 mm. The shape of the robot was also estimated using the approach described by Tully *et. al.* [3] for the sake of comparison.

The average error between the estimated shape and the ground-truth in both the experiments is tabulated in Table. 5.1. Fig. 5-5 shows the comparison between the



(a) Experiment 1 (b) Experiment 2

Figure 5-5: Blue-dotted curve is the ground-truth, green-thick curve is the estimated shape using proposed approach and red-thin curve is the shape estimate using the approach described in [3]



(a) Experiment 1 (b) Experiment 2

Figure 5-6: Comparison of variation in error between estimated position and ground-truth across the links. Green line is the variation using proposed approach and red dotted line is the variation using approach described by Tully *et. al.* [3]

Table 5.1: Comparison between estimated shape and ground-truth shape of the medical snake robot

	Experiment 1	
	proposed approach	approach in [3]
Standard deviation (in mm)	1.0167	3.6942
Average error (in mm)	4.7708	6.7434
Worst case error (in mm)	7.1578	13.8296
	Experiment 2	
	proposed approach	approach in [3]
Standard deviation (in mm)	3.702	8.368
Average error (in mm)	16.406	18.768
Worst case error (in mm)	22.032	29.444

ground-truth, the shape estimated by the proposed work and the shape estimated by Tully *et. al.* [3]. Note how the shape estimated by the proposed approach closely approximates the ground-truth. Fig. 5-6 shows the variation of the error between the ground-truth and the shape estimated for various links. Notice how the proposed method has low errors throughout, which is reflected in the low standard deviation as shown in the Table. 5.1. The improvement in the results obtained is attributed entirely to changing the space of the state vector to exponential coordinates.

5.1.4 Conclusion

Our algorithm makes use of sensing the tip position using an electro-magnetic sensor and motions models to predict and update the shape of the entire robot. An important contribution of this work has been describing the state at every instant, analytically in terms of the exponential coordinates and using these as state vector for the filtering to obtain more accurate estimates. Promising results have been obtained using the formulation described in this chapter demonstrating the capability of this approach to accurately filter the configuration of the robot in real-time.

The method described above can be extended to any system as long as the motion model can be obtained in closed-form in terms of exponential coordinates. Future work would involve sensing the motor-encoder readings and using that in the measurement update step to account for uncertainties arising from inaccurate motor inputs. Also more advanced models that can capture the interaction of the robot with deformable bodies is a subject of future work.

5.2 SCAR estimation using Lie algebra filter

In this section we build upon the SCAR-LSQ-CMU approach described in Chapter 3, where simultaneous registration and compliance estimation is done using a mini-batch of force-position measurements while assuming that the tool-tissue interactions are approximated by a linear spring-based stiffness model. In this section, we relax the assumptions of linearity in the tool-tissue interactions to account for palpation depth-

dependent changes in the stiffness, a commonly observed phenomena in deformable materials such as soft tissue [179]. We also utilize the experimental system presented in the previous work of our collaborators at Vanderbilt university [128], where an approach for stiffness and impedance exploration and autonomous motion in a deep cleft were investigated.

Furthermore, in this section, a pre-registration step is presented to estimate a prior for the filter which uses stiffness features segmented from a ground truth stiffness map, that can be generated either by ultrasound elastography or mechanical palpation. The pre-registration gives a good initial guess that helps in faster convergence of the filter as opposed to an uninformative prior. In addition, an unscented Kalman filtering approach is adopted with the hope of handling non linearities in process and observation model better than extended Kalman filters or iterated extended Kalman filters. The registration parameters are defined in the space of Lie algebra) exponential coordinates) as they have proved to be more accurate at estimating pose for localization [1] and shape estimation of medical snake robot as shown earlier in this Chapter.

5.2.1 Methodology

In this work, we first develop a structured framework to estimate the material properties of the soft body being probed. To do this, we discretize the surface of the soft body into a grid mesh and at each of the grid points, we fit a polynomial to describe the variation of the stiffness with respect to change in palpation depth. It is worth noting that while this ground truth variation of stiffness with respect to palpation depth is calculated, perfect registration between the soft body and its CAD model is assumed. A detailed description of this is provided in Section 5.2.2. For any unknown position of the organ with respect to the robot base, a pre-registration is performed to find a good initial registration estimate. This is done by scanning the surface at two different controlled force levels and estimating a rough stiffness contour map. Following that, the stiff regions are segmented from the contour stiffness map and the centroids of the stiff features are compared against the centroids of stiff features

on the ground truth stiffness map. The registration between the centroid points of the stiff features is obtained using Horn's method [4]. Section 5.2.3 describes the pre-registration procedure in detail.

The registration is carried out by using an unscented Kalman filtering framework. The registration estimate from the pre-registration step is used as the prior and the using the polynomial description for stiffness variation at grid points, the palpation force is modeled given the depth estimate at the point of palpation. The innovation step of the filter comprises of minimizing the difference between the modeled force and the measured force of palpation. Incorporating the raster palpation trajectory in the process model helps converge faster. The filtering is performed in the space of the Lie groups as developed by Hauberg *et. al.* [49], the procedure to do the same and its benefits are described in Section 5.2.4.

5.2.2 Generating the ground truth stiffness map

Prior work by Sanan *et. al.* [78] assumes a linear stiffness model for the palpated object where the stiffness did not vary with the palpation depth. In most practical applications however, this assumption is unlikely to hold due to the nonlinear material properties of biological tissues, which has been well established by Fung [180]. Capturing nonlinear material properties become crucially important when palpation is being carried out for a subsurface embedded object such as a tumor or an artery, which are located significantly below the organ surface, quite possibly beyond the linear approximation of the stiffness relating force with the palpation depth.

In order to characterize depth dependent stiffness in the palpated object, we perform the following steps (see Fig. 5-7):

1. The surface is raster scanned with the 3 DOF cartesian robot (See Section IV for further details). The point cloud of the robot tip $\boldsymbol{p}_i^R = (x_i, y_i, z_i)$, the contact force $\boldsymbol{f}_i^R$ and the local surface normal $\boldsymbol{n}_i^R$ at each point $\boldsymbol{p}_i^R$ are acquired. The superscript R refers to entities in the robot's reference frame (refer to Table 3.1 to understand the notations). This process is repeated for $i = 1 \dots q$, where q

is the total number of constant force scans.

2. The raw data $\boldsymbol{f}_i^R$, $\boldsymbol{n}_i^R$, and the z-coordinate of the point cloud z_i are fitted to a surface at pre-specified X-Y planar grid points (g_x, g_y) using a linear interpolation scheme. This leads to the following fitted data $\boldsymbol{n}_i^G$, $\boldsymbol{f}_i^G$ and $\boldsymbol{p}_i^G = (g_x, g_y, \hat{z}_i)$. The superscript G refers to entities in the frame of the grid.

3. At each grid location (g_x, g_y), the stiffness along the local surface normal is computed using a backward difference scheme, and is given as:

$$s_i(g_x, g_y) = -\frac{{\boldsymbol{n}_{i-1}^G}^T(\boldsymbol{f}_i^G - \boldsymbol{f}_{i-1}^G)}{{\boldsymbol{n}_{i-1}^G}^T(\boldsymbol{p}_i^G - \boldsymbol{p}_{i-1}^G)} \tag{5.27}$$

   The negative sign in the right hand side of Eqn. 5.27 accounts for the fact that the force acting on the palpated object is opposite to the reaction force on the force sensor measuring $\boldsymbol{f}_i^R$. Representative stiffness maps are shown in Fig. 5-8.

4. At each grid location (g_x, g_y), the palpation depth d_i, computed relative to the first scan corresponding to $i = 1$, and is given by:

$$d_i = \boldsymbol{n}_1^{GT}(\boldsymbol{p}_i^G - \boldsymbol{p}_1^G) \tag{5.28}$$

5. At each grid point (g_x, g_y), s_i is regressed on d_i using the quadratic relation:

$$s_i(d_i) = ad_i^2 + bd_i \tag{5.29}$$

   where a and b are regression coefficients.

5.2.3 Pre-registration

The pre-registration step is carried out to find a rough initial guess for the registration, which can then be used as a prior in the filtering process described in Section 5.2.4 for estimating the registration. In prior work by Sanan *et. al.* [78], no pre-registration

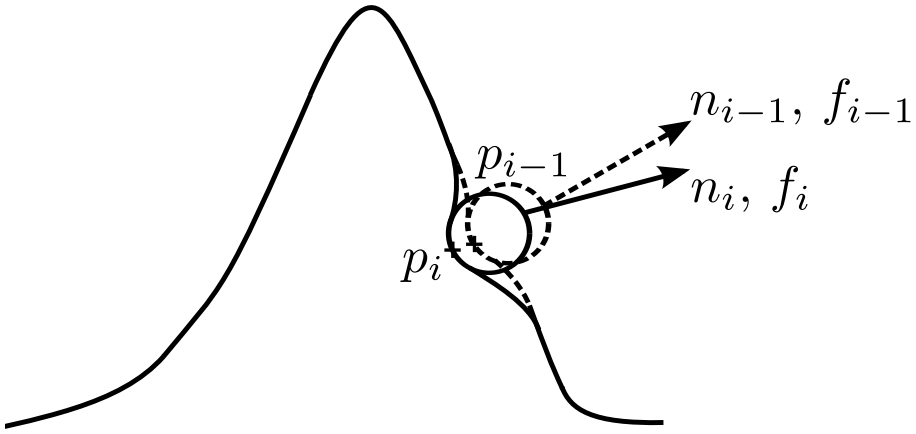


Figure 5-7: Schematic describing location of palpating points and the local surface normal at consecutive force scans.

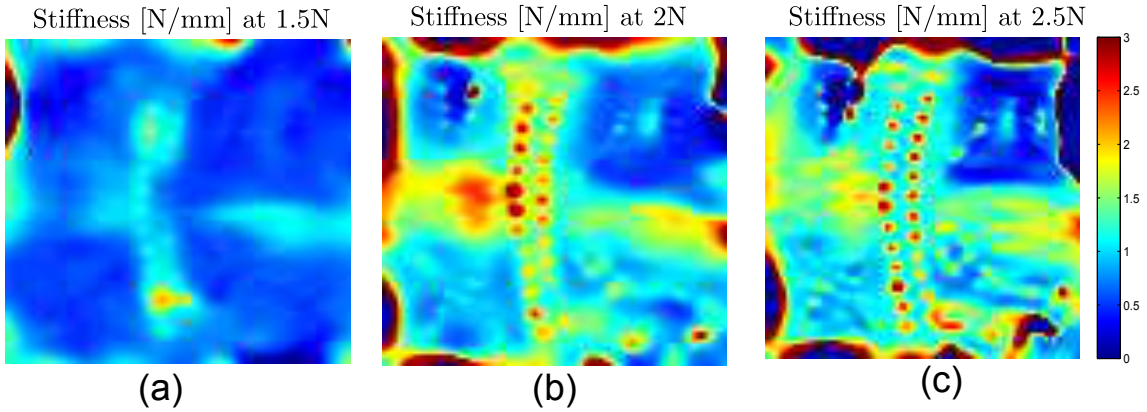
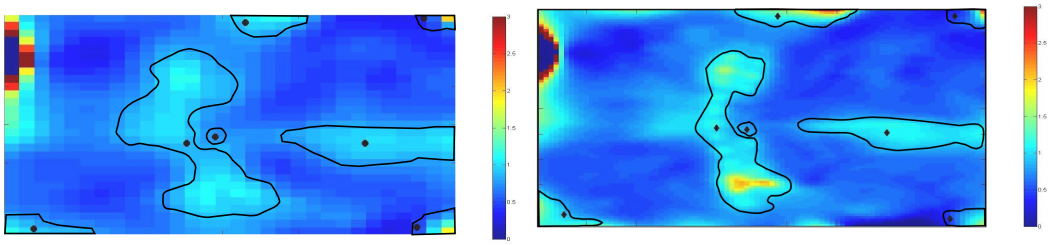


Figure 5-8: Representative stiffness distributions obtained at (a) $f_2 = 1.5N$, (b) $f_3 = 2.0N$ and (c) $f_4 = 2.5N$.

was done and the initial guess was arbitrarily chosen. However, it will be shown in Section 6.3 that choosing a reasonable initial guess from a pre-registration step provides more accurate registration estimate. To perform pre-registration, the organ is palpated at a very low force value. A low force value $\boldsymbol{f}_1$ would deform the surface very little and the tip positions obtained would closely emulate the points on the surface of the organ model. Following this, the organ is palpated at a higher force value $\boldsymbol{f}_2$ and the tip positions are recorded. The depth of palpation is estimated from the two palpation experiments performed and the stiffness is estimated assuming a linear stiffness model. We justify such an assumption by acknowledging the fact that in the pre-registration step we are not interested in computing the exact stiffness, but instead we wish to identify the approximate locations of relatively stiff regions. The location of the stiff regions are used as anatomical fiducials to register to the stiff regions in the ground truth stiffness map.



(a) Stiffness map obtained by palpating at arbitrary locations with 1.5N force. Centroids of locally stiff regions are shown by circular markers.

(b) Ground truth stiffness map for 1.5N force. Centroids of locally stiff regions are shown by diamond shaped markers.

Figure 5-9: Locally stiff regions are segmented from the stiffness map and centroid of the segmented regions are computed.

The relative-stiffness contour map of the surface of the organ that is obtained, shows contours of varying stiffness with varying color gradients as shown in Fig. 5-9(a). While the stiffness map in Fig. 5-9(a) looks like a blurred out version of the ground truth stiffness map shown in Fig. 5-9(b), the stiff regions still appear to be at the same relative position. Thus we compare the location of the centroids of the stiff regions to infer the transformation. The regions corresponding to higher stiffness are segmented as shown in Fig. 5-9. Further the centroids of the segmented regions

are computed. The coordinates of the centroids computed in the image frame can be readily transformed to coordinates in the frame of the robot base. Similarly the centroids of stiff regions are obtained from the ground truth stiffness map described in Section 5.2.2 corresponding to force $\boldsymbol{f}_2$. The coordinates of the centroids in the ground truth stiffness map can be readily transformed to the frame of the organ model. The registration between these two sets of centroids can be obtained using Horn's method [4].

5.2.4 Registration

Registration process involves estimating the homogeneous transformation matrix $\boldsymbol{T} \in SE(3)$, that relates the reference frame fixed to the base of the robot to the reference frame attached to the model of the organ of interest. In this work, a filtering approach is followed for estimating the registration. Unlike the prior work from our group [47, 78] where the state vector of the filter contains Euler angles and Cartesian coordinates that parameterize $\boldsymbol{T}$, in this work the registration is carried out directly in the space of the Lie group by using a state matrix: $\boldsymbol{T}$. The underlying Lie algebra of the space is used to derive the equations of the filter as shown in [49]. Filtering in the tangent space of the group provides more accurate estimation of the state as shown in [1, 49], especially in the presence of high uncertainties in the process or measurement models. In this work, we therefore adapt this approach of filtering in the space of the Lie group, instead of choosing a Euclidean parameter space.

The state matrix is defined as:

$$\boldsymbol{X}_k = \boldsymbol{T}, \quad \text{where } \boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{pmatrix}, \tag{5.30}$$

where $\boldsymbol{R} \in SO(3)$ is the rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector. We use a UKF in this work as it is known to handle non-linearities in the process and measurement models better [181]. The unscented filtering framework involves computing intermediate states called 'sigma points' which are propagated through the non-linear

process and measurement models of the system.

Sigma point computation

Consider the state $\boldsymbol{X} \in SE(3)$ and covariance $\boldsymbol{P}$ expressed in the basis of the tangent space at $\boldsymbol{X}$. The tangent space $se(3)$ can be described using a six dimensional basis commonly referred to as exponential coordinates [87]. Let $\boldsymbol{\sigma}_T = \boldsymbol{\sigma}^0, \ldots, \boldsymbol{\sigma}^{12}$ be the sigma points of the covariance $\boldsymbol{P}$ calculated in the Euclidean tangent space:

$$\boldsymbol{\sigma}^0 = \boldsymbol{x}, \tag{5.31}$$

$$\boldsymbol{\sigma}^m = \boldsymbol{x} + \left(\sqrt{(M+\lambda)P}\right)_m, \quad m = 1, \ldots, M \tag{5.32}$$

$$\boldsymbol{\sigma}^m = \boldsymbol{x} - \left(\sqrt{(M+\lambda)P}\right)_m, \quad m = M+1, \ldots, 2M \tag{5.33}$$

where $M = 6$ the dimension of the tangent space of $SE(3)$, $\hat{\boldsymbol{x}} = \log \boldsymbol{X}$ are the exponential coordinates that describe $\boldsymbol{X}$, $(\sqrt{\cdot})_m$ denotes m^{th} column of the Cholesky decomposition and λ is a parameter for controlling the distance between the sigma points and the mean. The sigma points obtained are then projected to the manifold describing $SE(3)$ using the matrix exponential: $\boldsymbol{\sigma}^m_{SE(3)} = \exp \boldsymbol{\sigma^m}$, where $m = 0, \ldots, 12$.

Process and measurement models

The filtering process comprises of two main steps: prediction and update. In the prediction step, a process model is used to predict the future state of the system given it current estimate and in the update step, sensor data is used to correct the state estimate. In this work, the process model is designed to be static i.e., $\boldsymbol{X}_{k|k-1} = \boldsymbol{X}_{k-1|k-1}$ and $\boldsymbol{P}_{k|k-1} = \boldsymbol{P}_{k-1|k-1}$. The control input for the system is the estimated tip position of the robot. The tip position is estimated using another UKF as described in Section 5.2.4. The controlled force applied to the organ during the palpation serves as the measurement. The measurement model for this system is:

$$h(\boldsymbol{X}_k) = \delta_k S(\boldsymbol{p}_c, \delta_k), \quad \text{where } \delta_k = \boldsymbol{n}^T(\boldsymbol{p}_c - \boldsymbol{p}_c^{tip}), \tag{5.34}$$

where the subscript c denotes quantities in the frame of the organ model and $S(\boldsymbol{p}_c, \delta_k)$ is a function that returns the stiffness value associated with a point $\boldsymbol{p}_c$ on the organ model when palpated to a depth δ_k, as described in Section 5.2.2. The point $\boldsymbol{p}_c^{tip} = \boldsymbol{R}_k \boldsymbol{p}_r^{tip} + \boldsymbol{t}_k$, where $\boldsymbol{X}_k = \begin{pmatrix} \boldsymbol{R}_k & \boldsymbol{t_k} \\ \boldsymbol{0} & 1 \end{pmatrix}$ and $\boldsymbol{p}_r^{tip}$ is in the frame of the robot base as estimated is Section 5.2.4. In Eq. 5.34, the surface normal $\boldsymbol{n}$ is obtained by finding the normal associated with the point $\boldsymbol{p}_c$ on the organ model. The measurement function Eq. 5.34 is evaluated at the sigma points obtained earlier:

$$y_k = \sum_{m=0}^{12} w_m h(\boldsymbol{\sigma}_{SE(3)}), \tag{5.35}$$

where the weights w_i are defined as:

$$w_0 = \lambda/(\lambda + M), \tag{5.36}$$

$$w_m = 1/2(\lambda + M), \quad m = 1, \ldots, 2M, \tag{5.37}$$

where the dimentionality of the space, $M = 6$.

Update step

The update step uses the sensor data z_k to update the mean and variance and obtain $\boldsymbol{X}_{k|k}$ and $\boldsymbol{P}_{k|k}$. The covariance and cross-covariance required to compute the Kalman gain are:

$$\boldsymbol{P}_{yy} = \sum_{m=0}^{1} 2w_m (h(\boldsymbol{\sigma}_{SE(3)}^m) - y_k)(h(\boldsymbol{\sigma}_{SE(3)}^m) - y_k)^T,$$

$$\boldsymbol{P}_{xy} = \sum_{m=0}^{1} 2w_m (\boldsymbol{\sigma}^m - \boldsymbol{x}_{k|k-1})(h(\boldsymbol{\sigma}_{SE(3)}^m) - y_k)^T.$$

The Kalman gain can be obtained as $\boldsymbol{K} = \boldsymbol{P}_{xy}\boldsymbol{P}_{yy}^{-1}$. The sensor measures the normal force applied to the organ and thus we have $z_k = F_{\boldsymbol{n}}$. The estimate of the state is

updated as follows:

$$\boldsymbol{X}_{k|k} = exp(\boldsymbol{x}_{k|k-1} + \hat{\boldsymbol{K}}(z_k - y_k)). \tag{5.38}$$

The covariance is updated as follows:

$$\boldsymbol{P}_{k|k} = \boldsymbol{P}_{k|k-1} + \boldsymbol{K}\boldsymbol{P}_{yy}\boldsymbol{K}^T. \tag{5.39}$$

Estimation of tip position

The control input to the UKF presented above is $\boldsymbol{p}_r^{tip}$, which is the position of the tip of the robot in the frame of the base of the robot. At every iteration of the UKF, $\boldsymbol{p}_r^{tip}$ is estimated using another UKF. The state vector of this second UKF is $\boldsymbol{q}_k = \boldsymbol{p}_r^{tip} \in \mathbb{R}^3$. The process model for the filter uses a constant velocity motion model:

$$\boldsymbol{q}_{k|k-1} = \boldsymbol{q}_{k-1|k-1} + \boldsymbol{v}\Delta t,$$
$$\boldsymbol{Q}_{k|k-1} = \boldsymbol{Q}_{k-1|k-1} + \boldsymbol{N},$$

where $\boldsymbol{v} = (\boldsymbol{q}_{k-1|k-1} - \boldsymbol{q}_{k-2|k-2})/\Delta t$ and Δt is the time elapsed between two successive filtering loops and N is the uncertainty in the process model that helps take care of unmodeled scenarios such as change of direction of motion, accuracy of the robot etc. The measurement model for this filter is:

$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{q},$$

The sensor measurement is the tip position: $\boldsymbol{z}_k = \boldsymbol{p}^s$. The sigma points and weights are computed using Eq. 5.31 and Eq. 5.36 respectively, with the dimensionality of the state vector $M = 3$. Since the state vector belongs to the Euclidean space, standard equations for prediction and update for the UKF [181] are used. The state $\boldsymbol{q}$ is continuously updated and whenever the UKF that estimates the registration requires a control input, the latest estimate of $\boldsymbol{q}$ is used.

5.2.5 Experimental setup

To demonstrate the validity of the proposed approach, palpation experiments at various constant forces were carried out, as outlined in Section 5.2.2. The experiments were carried out by our collaborators at Vanderbilt university. The experimental setup consisted of a 3-DOFs cartesian robot equipped with a force/torque sensor (ATI Gamma F/T Transducer, S/N: FT8578) with an attached spherical probe 12.7 mm in diameter (see Fig 5-10. Centralized computed torque control for the Cartesian robot was implemented in a real-time system running at 1 kHz. Hybrid-force control was used for raster positioning a silicone tissue phantom (M-F Manufacturing), wherein force control was applied in along the local surface normal computed from the F/T sensor, and position control was applied along the orthogonal direction. Further details on the control architecture can be found in [128]. This is the same experimental setup what was used in Chapter 3.

Five consecutive force-controlled scans were carried out from 1.0 N to 3.0 N at intervals of 0.5 N. We observed that at higher forces, there was considerable distortion in the phantom tissue, due to which data acquired at these forces were not considered in the registration process.

5.2.6 Results for SCAR with Lie algebra filter

As a first step, the ground truth stiffness maps were computed for various forces as described in Section 5.2.2. The stiffness maps for three sample forces are shown in Fig. 5-8. Since the stiffness computations are interpolated over the surface of the organ by using a grid mesh of points taken along a plane and projected to the surface, the estimates are poor at the regions where the surface normals undergo a quick change in orientations. As a result we omit points belonging to such regions during our computations due to high uncertainties in the estimated values at those points. Fig. 5-11(a) shows a contour plot of the variation in orientation of the surface normals and in Fig. 5-11(b), the points in the light colored regions are not considered for stiffness computation in the UKF.

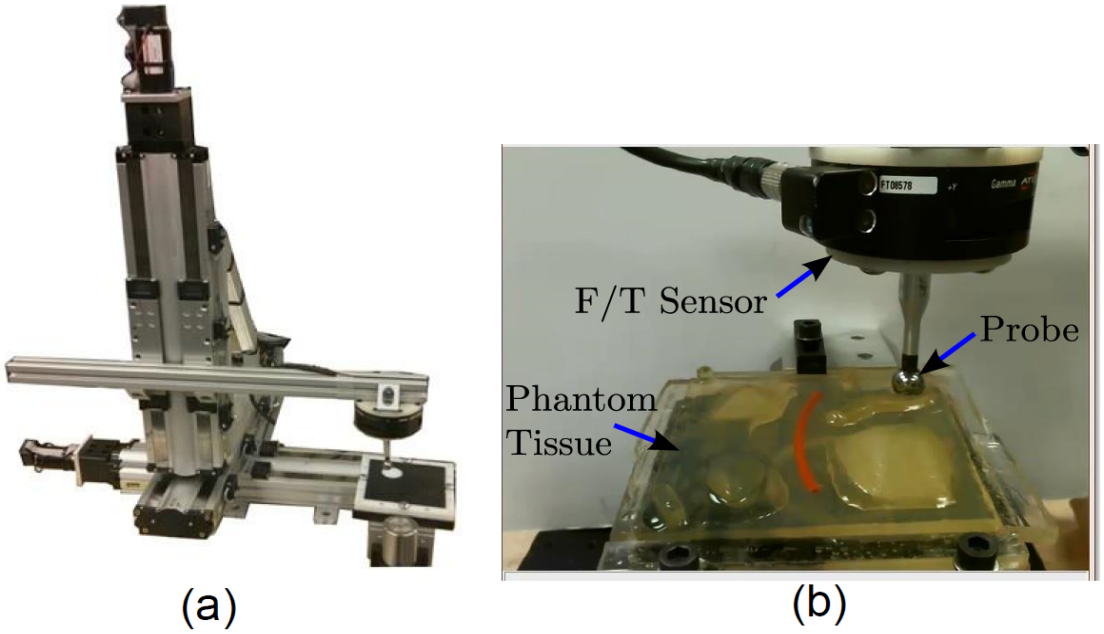
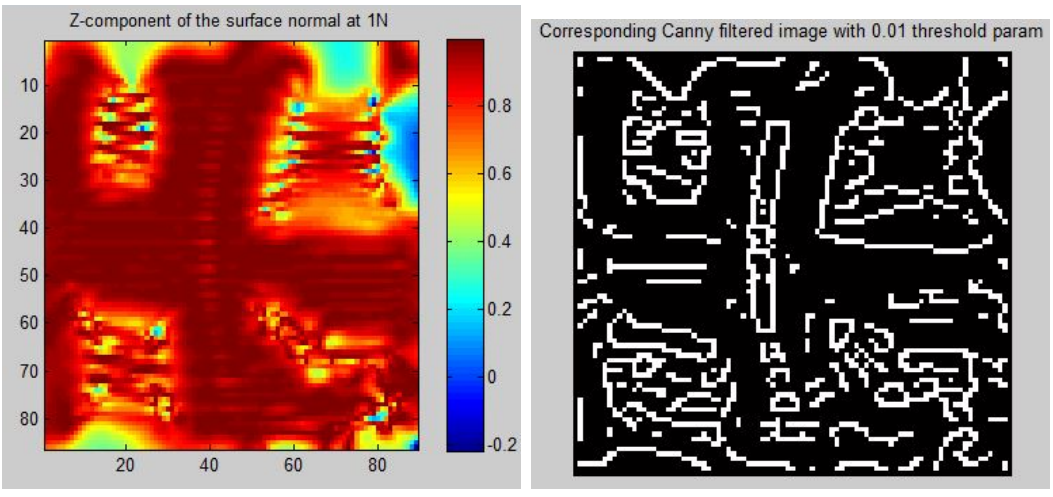


Figure 5-10: (a) 3-DOF Cartesian Robot at Vanderbilt University and (b) Experimental setup with the probe palpating the phantom tissue.



(a) Contour plot of the variation in surface normal over the surface of the model

(b) The white regions show high variation in surface normal orientation

Figure 5-11:

The location of the organ with respect to the base of the robot was carefully measured to provide a ground truth registration, in order to measure how good our estimated registration is. The ground truth registration is:

$$\boldsymbol{T} = \begin{pmatrix} 0.9996 & 0.0276 & 0.0101 & -34.7452 \\ -0.0278 & 0.9995 & 0.0171 & -7.2757 \\ -0.0096 & -0.0174 & 0.9998 & 16.3866 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}. \tag{5.40}$$

The surface is first palpated at two force levels, 1N and 1.5N at arbitrary locations. The tip location is sensed in both these experiments and a stiffness map is generated from this data following the approach shown in Section 5.2.2. The stiffness map generated is as shown in Fig. 5-9. Upon performing a pre-registration as shown in Section 5.2.3 we obtain the following transformation matrix:

$$\boldsymbol{T} = \begin{pmatrix} 0.9990 & 0.0436 & 0.0104 & -34.8915 \\ -0.0438 & 0.9989 & 0.0169 & -6.5140 \\ -0.0096 & -0.0174 & 0.9998 & 16.3866 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}. \tag{5.41}$$

Having performed the pre-registration, the surface was raster scanned as shown in Fig. 5-12 with a constant force of 1.5N.

Palpation points were sampled at regular time intervals along the trajectory. Fig. 5-13(a) shows the coordinates of the palpation points against the surface of the organ. The UKF was used to estimate the registration using the prior from pre-registration step. The UKF successfully registers the points to the surface of the

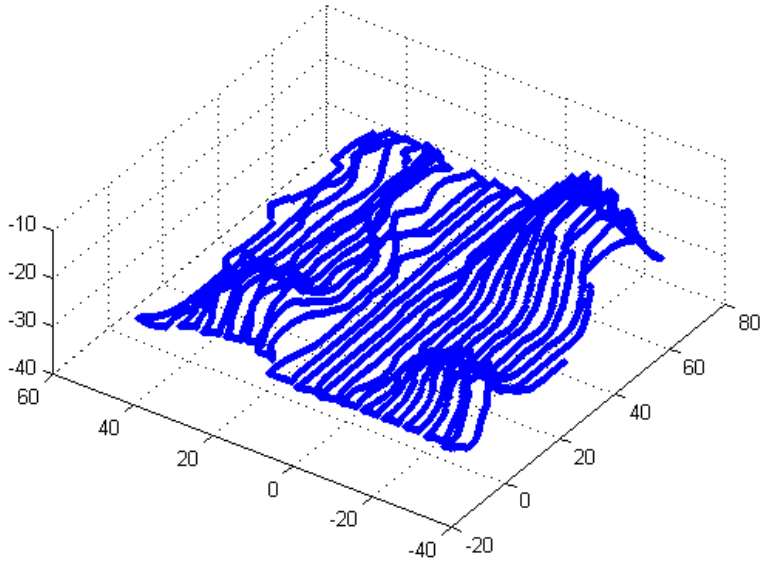


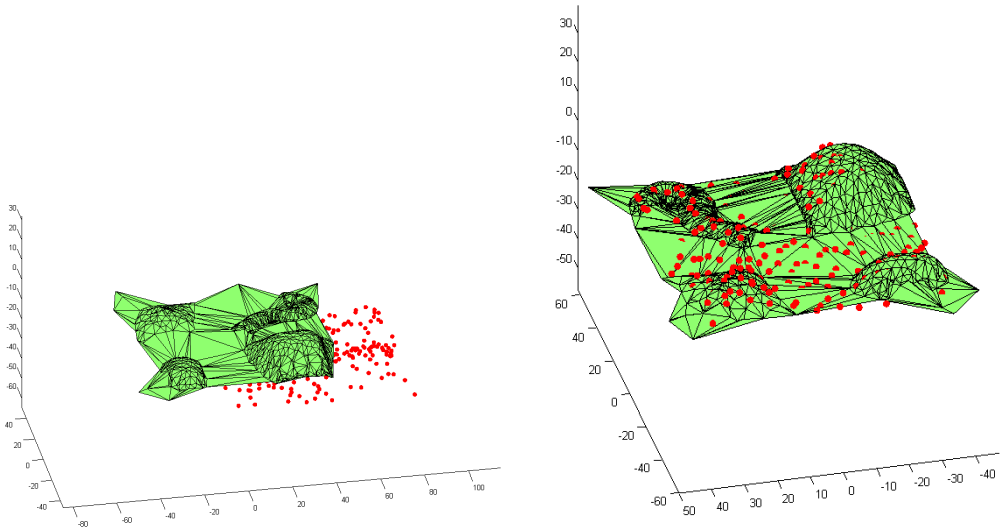
Figure 5-12: Raster path that was used to palpate the silicon model

model as shown in Fig. 5-13. The final registration as estimated by the filter was:

$$\boldsymbol{T} = \begin{pmatrix} 0.9989 & -0.0316 & 0.0333 & -34.5227 \\ 0.0320 & 0.9994 & -0.0111 & -8.5058 \\ -0.0329 & 0.0122 & 0.9994 & 16.7827 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix},$$

which is very close to the ground truth. When the registration estimation was performed without using the palpation trajectory model, the estimates for tip position took longer time to converge.

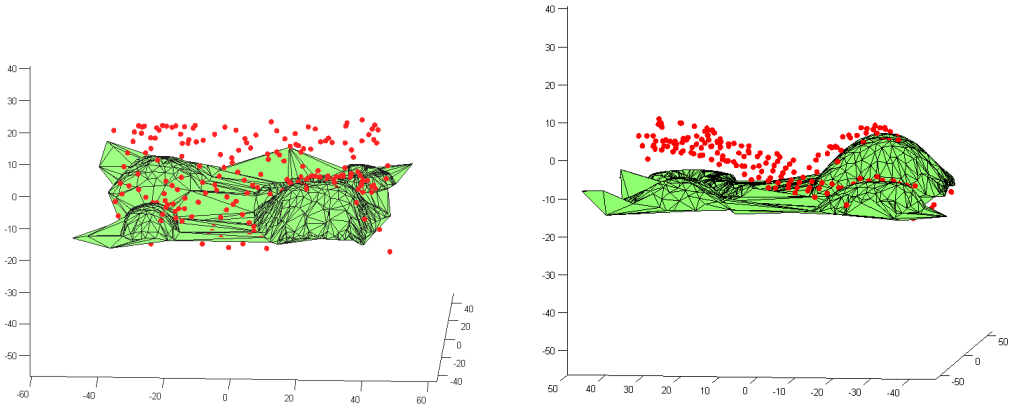
The estimation procedure when performed with an arbitrary initial guess, yields inferior results. A sample result for a failed registration is as shown in Fig. 5-14. This example demonstrates the importance of the pre-registration step. The registration was also estimated using an IEKF instead of UKF and the results obtained were found to be very similar.



(a) The red dots represent the palpation points in the frame of the organ model before registration

(b) The red dots represent the palpation points in the frame of the organ model after registration

Figure 5-13: Example of successful registration using the initial guess for the UKF from pre-registration step.



(a) The red dots represent the palpation points in the frame of the organ model before registration

(b) The red dots represent the palpation points in the frame of the organ model after registration

Figure 5-14: Example of bad registration when the intial guess for UKF is not chosen from the pre-registration step.

5.2.7 Conclusion

We have developed a probabilistic approach based on unscented Kalman filtering to register a deformable object to a surgical tool using mechanical stiffness information obtained by palpating the object. We assumed a realistic deformation dependent stiffness model obtained by raster-scanning the object at varying forces along the local surface normal of the object using a 3-DOFs Cartesian robot. Subsequently, the stiffness distribution at a given scan force was used to pre-register the object to the robot tip. In addition, a pre-defined palpation trajectory was used to specify the process model of the filtering algorithm instead of uniformative distribution of the palpation path.

To the best of the our knowledge, this chapter presents for the first time, a mechanical palpation based pre-registration step to provide a good initial guess to the registration process. One of the key findings is that a pre-registration step significantly improves registration accuracy. The implication of this result is that registration of the surgical tool inside the human anatomy improves if an accurate pre-operative elastic map of the anatomy of interest is provided prior to the surgical task . We also observed that absence of a pre-defined trajectory leads to inaccurate registration. These findings encourage further research in proper selection of the palpation trajectory that leads to the most optimal registration.

In this work, we did not consider global deformations in the deformation during palpation. This is not an unlikely scenario, since anatomical structures are typically geometrically unconstrained and could move globally during the palpation process. In the future, we will focus on accounting for global deformations in the registration process.

5.3 Contribution and Discussions

The contributions from this chapter include:

1. Derivation of closed form analytical expressions for the motion model to esti-

mate the shape and pose of a medical snake robot, defined in terms of exponential coordinates.

2. Using a realistic stiffness model for SCAR, for improved estimates.

3. A pre-registration step in SCAR using prior stiffness information.

4. An unscented filter for pose estimation using Lie algebra to parameterize pose.

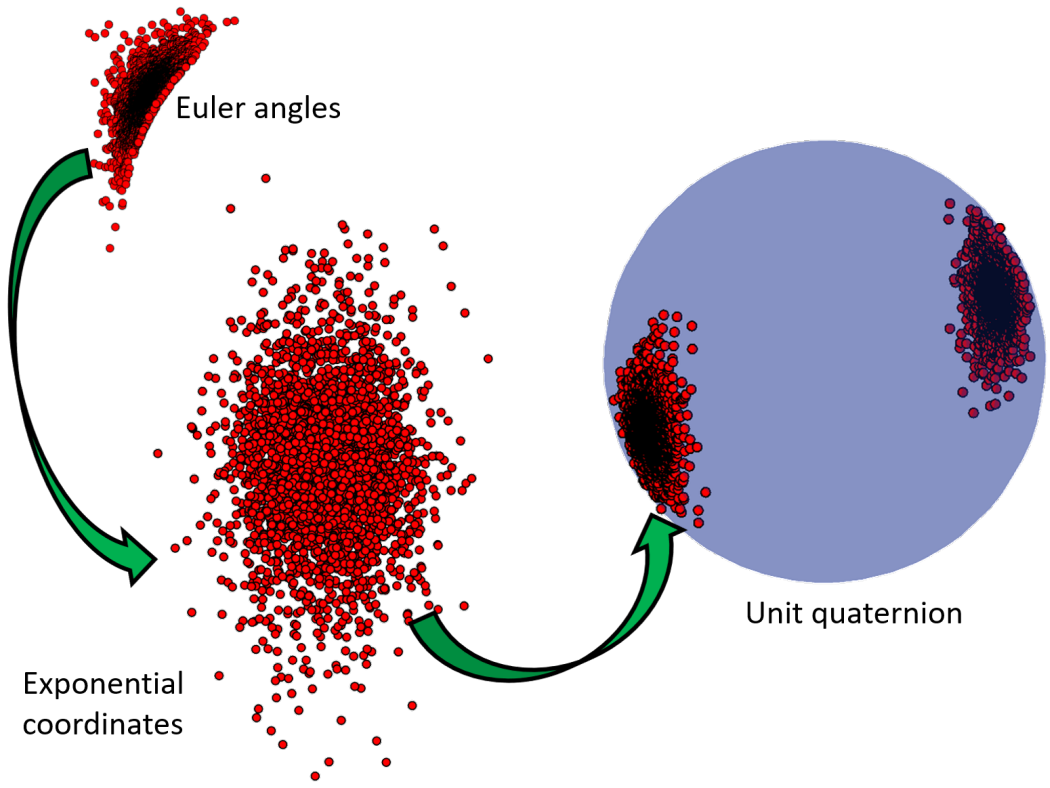


Figure 5-15: When particles obtained from a banana shaped distribution in the space of Euler angles (top left) are transformed to the space of exponential coordinates they are distributed in the form of a Gaussian distribution (center). When particles obtained from a Gaussian distribution in the space of exponential coordinates are transformed to the space of unit quaternions, they are distributed in the form of a Bingham distribution (right).

An important learning from this Chapter was that using exponential coordinates to parameterize pose and perform filtering produces accurate pose estimates, as op-

posed to using Cartesian coordinates and Euler angles. However, it must be noted that even for the case of static pose estimation (where there is no process model), the update model is nonlinear and requires linearization (as in the case of medical snake robot's shape estimation) or deterministic sampling approximations (as in the case of SCAR); which can be susceptible to high initialization errors.

However, we observe that a Gaussian in the space of exponential coordinates map to a Bingham distribution in the space of unit quaternions (as shown in Fig. 5-15). In the next chapter we show how we make use of this mapping to derive the equations of a linear filter that uses Bingham distribution to model the uncertainties in rotation parameters. We show that such a formulation is not only more accurate but also automatically handles the constraints imposed by the chosen parameterization.

5.4 Published Work

Material from this chapter has appeared in the following publications

1. R Arun Srivatsan, Matthew Travers, and Howie Choset, "Using Lie algebra for shape estimation of medical snake robots, in the proceedings of the IEEE/ RSJ International Conference on Intelligent Robots and Systems, Chicago, USA, 2014, pg 3483-3488.

2. R Arun Srivatsan, Rajarshi Roy, Long Wang, Nabil Simaan, and Howie Choset, "Registering Surgical Tool to a Soft Body using Mechanical Palpation", Tech. report CMU-RI-TR-15-13, Robotics Institute, Carnegie Mellon University, June, 2015.

Chapter 6

Bingham Filter for Pose Estimation

Most of the prior work on online pose estimation linearize the non-linear measurement model. This results in inaccurate estimates especially when the initial pose estimate is erroneous. In order to address this issue, in Chapter 4, we used dual-quaternions and developed a linear Kalman filter which is robust to initial pose errors. Our prior work as well as the work of others ([154, 62, 63, 143, 182]), used unit-quaternions as the underlying space and Gaussians to model the uncertainty in their distribution. Gaussian distributions do not consider the structure of the underlying space, i.e, antipodal symmetry introduced by $\widetilde{\boldsymbol{q}} = -\widetilde{\boldsymbol{q}}$ ([81]).

Gaussian distributions can be used to model uncertainty, but are largely limited to linear spaces and states that are unimodal in distribution. While much work has been done to counteract this assumption, the bottom line is that linearizations are still approximations. We believe that in order to obtain optimal state estimates, it is critical to model uncertainties using the appropriate distributions that exploit the fundamental structure of the parameter space. This work introduces an online pose estimation method that uses a Bingham distribution and a Gaussian distribution to robustly and accurately estimate the rotation and translation respectively.

The Bingham distribution is defined on a unit hypersphere and captures the antipodal symmetry of the distribution of unit-quaternions ([57]) (see Fig. 6-1). When

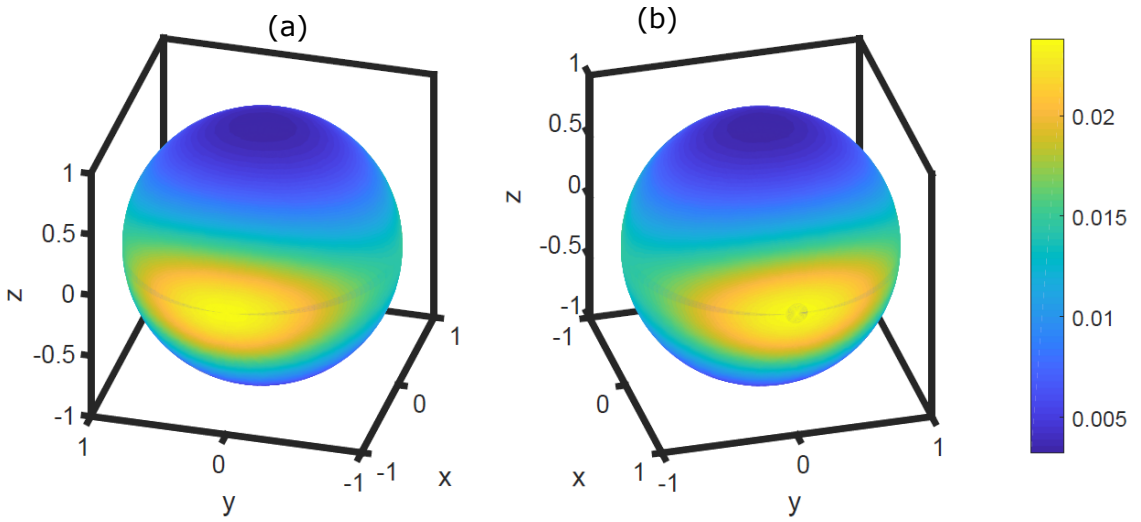


Figure 6-1: A 3D Bingham distribution: $c = \frac{1}{N}\exp(\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{Z}\boldsymbol{M}^T\boldsymbol{v})$, where $\boldsymbol{M} = \boldsymbol{I}_{3\times3}$, $\boldsymbol{Z} = diag(0, -0.5, -2)$, and $\boldsymbol{v} = (x, y, z)^T$, $\boldsymbol{v}^T\boldsymbol{v} = 1$. The colors on the sphere show the probability value. (a) Shows the mode at $\boldsymbol{v} = (1, 0, 0)^T$, (b) shows the mode at $\boldsymbol{v} = (-1, 0, 0)^T$. More details can be obtained from Sec. 2.5.

compared to prior methods, the use of the Bingham distribution results in a formulation that is accurate and has shorter computation time, because there is no normalization step or projection onto a hyper-sphere.

Another advantage of our approach compared to the UKF-based pose estimation method of Moghari *et. al.* [5] and EKF-based pose estimation method of Pennec *et. al.* [6]) is the ability to update the pose not only using point measurements, but also using surface-normal and pose measurements as well as simultaneous multiple measurements (as obtained from a stereo camera or lidar).

Inspired by Horn [4], this work estimates pose by decoupling orientation from translation estimation. The method uses a Bingham distribution-based filter (BF) for orientation estimation and a Kalman filter for translation estimation. While there has been some recent work on using the BF for orientation estimation ([81], [54]), there are some key differences compared to our approach. Firstly, prior work assumes that the state and measurements both are unit quaternions. Secondly, prior works deal with non-linear measurement models [183], hence requiring approximations introduced by linearization or deterministic sampling such as the works of Steinbring *et. al.* [184] and Gilitschenski *et. al.* [54]. This results in computation of the normalization constant

which is known to be expensive ([13, 81]). On the other hand, our approach does not limit measurements to be unit quaternions and bypasses the computation of normalization constant by using a linear measurement model. The computation of normalization constant would still be required to find the covariance of the orientation parameters, which is not part of the algorithm itself.

Faugeras *et. al.* [60] and Walker *et. al.* [61] have also derived linear models for pose estimation. However, they estimate pose in a deterministic setting and do not provide any guidelines for estimating the uncertainties associated with pose. Following Chaukroun *et. al.* [143] and the approach shown in Chapter 4, in this work we use results from stochastic filtering theory to derive exact expressions for the pose-uncertainties.

We first derive the theory for applications where the data association between the model and the sensor measurements is known. Following this, we extend the approach to applications with unknown data association. To deal with unknown data association, we use the popular k-d tree search and principal direction tree (PD-tree) search. We also introduce a new look-up table-based approach for fast computations when the data association is unknown. We show that the computational time taken by this approach is a couple of orders of magnitude less than tree search. We take advantage of the reduced computational time to recursively run the filter for dynamic pose estimation.

6.1 Related Work

6.1.1 Batch Processing Approaches

Pose estimation has been of interest for a long time in the robotics literature. Much of the early literature deals with collecting all sensor measurements and processing them offline in a batch to estimate the pose. Horn [4] developed a least squares implementation for pose estimation with known point correspondence. Besl and McKay [11] introduced the iterative closest point (ICP), which extends Horn's methods for unknown

point correspondence by iteratively estimating point correspondence and performing least squares optimization. Several variants of the ICP have been developed (refer to the review on registration methods by [12]). Estepar *et. al.* [21], Segal *et. al.* [22] and Billings *et. al.* [23] further generalized the ICP by incorporating measurement noise uncertainties. A more detailed discussion on these methods can be obtained in Chapter 9.

Orientation data (surface-normals) has been used in addition to point data for registration in prior works. The approach of Pulli [28] uses surface-normal information to filter out measurements during the correspondence stage. Munch*et. al.* [29] use point and surface-normal measurements in both the correspondence and minimization step. Billings *et. al.* [25] have developed iterative most likely oriented point (IMLOP), a probabilistic framework to estimate pose using surface-normal and position measurements, while incorporating measurement uncertainty in both the correspondence and minimization step. Billings *et. al.* [185] have more recently developed a generalized IMLOP (GIMLOP) to deal with anisotropic noise in points and surface-normals.

In addition to point and surface-normal measurements there have also been several approaches developed for pose estimation using pose measurements for applications such as hand-eye calibration. The works of Tsai and Lenz [30], Horaud and Dornaika [31] and Daniilidis [32] formulate hand-eye calibration as a least squares estimation problem assuming complete knowledge of measurement correspondence. More recently, Ackerman *et. al.* [33] estimate the pose with unknown correspondence.

6.1.2 Probabilistic Sequential Estimation

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

when to stop collecting measurements

Gaussian Filtering Approaches

Several sequential estimation methods are based on Kalman filters, which model the states and measurements using Gaussian distributions [6, 5, 10, 49, 101]. Kalman filters by construction provide optimal state estimates when the process and measurement models are linear and the states and measurements are Gaussian distributed [51]. Pose estimation, however, is inherently a non-linear problem, and hence linear Kalman filters produce poor estimates [6, 49]. Several variants of the Kalman filter have been introduced to handle the non-linearity. EKF-based filters perform first-order linear approximations of the non-linear models and produce estimates which are known to diverge in the presence of high initial estimation errors [5]. UKF-based methods do not linearize the models but instead utilize evaluation at multiple points, which can be expensive for a high-dimensional system such as $SE(3)$ [5, 49]. In addition, UKF-based methods require tuning a number of parameters, which can be unintuitive.

Non-Gaussian Filtering Approaches

There has been some recent work in robotics towards the use of alternative distributions to model the noise on rotations for pose estimation problems. For example, Langevin distributions have been used for pose estimation by Carlone *et. al.* [52] and Rosen *et. al.* [53]. Gilitschenski *et. al.* [54] have recently developed a Bingham distribution-based recursive filtering approach for orientation estimation. Glover *et. al.* [13] use Bingham distribution to describe the orientation features, while Gilitschenski *et. al.* [55] use this distribution for planar pose estimation. Our work takes inspiration from these works for modeling the uncertainty in the orientation using Bingham distribution. Post the publication of our work, Fan *et. al.* [56] developed a variation of that DQF developed in Chapter 4, by using a constrained Kalman filter with dual quaternions as state vector.

The use of Bingham distribution to model uncertainties in rotation parameters is a very valuable tool that has been largely under-utilized by the robotics community,

as also noted by Glover *et. al.* [13]. One of the important reasons for this, is the difficulty in computing the normalization constant as well as performing expensive convolution operation over the distributions [57].

Alternate Parameterizations for Filtering

Prior work also has looked at several parameterizations of $SE(3)$ that would improve the performance of the filters. [49] confine the state variables over a known Riemannian manifold and use a UKF to estimate the pose. Quaternions are used to parametrize $SO(3)$ and the state is estimated using an EKF in [62, 101] and UKF in [63]. An IEKF with dual quaternions to parameterize the pose has been used by Goddard *et. al.* [64].

Linear Filtering Approach

In Chapter 4, we have developed a linear Kalman filter for pose estimation using dual quaternions and pairwise measurement update. While this method has been shown to be robust to errors in initial state estimate and sensor noise, it has a few drawbacks: (1) The uncertainty in the quaternions used for orientation estimate is modeled using Gaussians which do not consider the condition that $\widetilde{\boldsymbol{q}}$ and $-\widetilde{\boldsymbol{q}}$ represent the same rotation. (2) The filter by itself does not produce unit-quaternion estimates and hence after each estimate, a projection step is used to normalize the state. The projection step introduces an additional error which is not accounted for in the uncertainty estimate [186]. In addition, such a projection would have a large error if the estimated state had a near zero norm. (3) The approach only requires pairs of measurements per update. However, in many practical applications such as image registration, several ($\approx 10^4$) measurements are available for processing in each update step, and a pairwise update could be very inefficient and time consuming.

6.2 Problem Formulation

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

6.2.1 Position Measurements

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

$$\boldsymbol{a}_i = Rot(\widetilde{\boldsymbol{q}})\boldsymbol{b}_i + \boldsymbol{t}, \quad i = 1, \ldots, n, \tag{6.1}$$

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

Update Model

First consider the scenario where points in the sensor frame are obtained one at a time in a sequential manner, as typically observed in the case of robotic probing (see Chapter 3). Similar to Chapter 4, the equations for updating the pose estimate given a pair of measurements ($n = 2$), are derived. From Eq. 2.15, Eq. 6.1 can be rewritten as

$$\widetilde{\boldsymbol{a}}_1 = \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_1 \odot \widetilde{\boldsymbol{q}}^* + \widetilde{\boldsymbol{t}}, \tag{6.2}$$

$$\widetilde{\boldsymbol{a}}_2 = \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_2 \odot \widetilde{\boldsymbol{q}}^* + \widetilde{\boldsymbol{t}}, \tag{6.3}$$

where $\widetilde{\boldsymbol{q}}$ is as defined in Eq. 2.14 and $\widetilde{\boldsymbol{t}} = (0, \boldsymbol{t}^T)^T$. Subtracting Eq. 6.3 from Eq. 6.2,

$$\begin{aligned} \widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2 &= \widetilde{\boldsymbol{q}} \odot (\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2) \odot \widetilde{\boldsymbol{q}}^*, \\ \Rightarrow (\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2) \odot \widetilde{\boldsymbol{q}} &= \widetilde{\boldsymbol{q}} \odot (\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2), \end{aligned} \tag{6.4}$$

since $\widetilde{\boldsymbol{q}}$ is a unit-quaternion. Using matrix form of quaternion multiplication shown in Eq. 2.11, Eq. 6.4 can be rewritten as

$$\begin{aligned} \boldsymbol{F}_1(\widetilde{\boldsymbol{a}}_1 - \widetilde{\boldsymbol{a}}_2)\widetilde{\boldsymbol{q}} - \boldsymbol{F}_2(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2)\widetilde{\boldsymbol{q}} &= 0, \\ \Rightarrow \boldsymbol{H}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2)\widetilde{\boldsymbol{q}} &= \boldsymbol{0}, \quad \text{where} \end{aligned} \tag{6.5}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & -(\boldsymbol{a}_v - \boldsymbol{b}_v)^T \\ (\boldsymbol{a}_v - \boldsymbol{b}_v) & (\boldsymbol{a}_v + \boldsymbol{b}_v)^\times \end{bmatrix} \in \mathbb{R}^{4\times 4}, \tag{6.6}$$

$\boldsymbol{a}_v = \boldsymbol{a}_1 - \boldsymbol{a}_2$ and $\boldsymbol{b}_v = \boldsymbol{b}_1 - \boldsymbol{b}_2$. Notice that Eq. 6.5 is a linear equation in terms of $\widetilde{\boldsymbol{q}}$ and is independent of $\boldsymbol{t}$.

Adding Eq. 6.2 and Eq. 6.3,

$$\begin{aligned} \widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2 &= \widetilde{\boldsymbol{q}} \odot (\widetilde{\boldsymbol{b}}_1 + \widetilde{\boldsymbol{b}}_2) \odot \widetilde{\boldsymbol{q}}^* + 2\widetilde{\boldsymbol{t}}, \\ \Rightarrow \widetilde{\boldsymbol{t}} &= \frac{(\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2) - \widetilde{\boldsymbol{q}} \odot (\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2) \odot \widetilde{\boldsymbol{q}}^*}{2}. \end{aligned} \tag{6.7}$$

Eq. 6.5 and Eq. 6.7 were derived in Chapter 4 using dual quaternions, however, no geometrical intuition was provided. Fig. 6-2 provides the geometrical intuition behind the decoupled estimation of $\widetilde{\boldsymbol{q}}$ and $\boldsymbol{t}$. Estimating the pose between $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ can be reduced to first estimating the orientation of vectors $\boldsymbol{a}_v^{ij}$ and $\boldsymbol{b}_v^{ij}$ and then estimating the translation between the centroids of the points. A similar idea is commonly used by Horn [4]. A key difference is that instead of forming vectors $\boldsymbol{a}_v = \boldsymbol{a}_1 - \boldsymbol{a}_2$ and $\boldsymbol{b}_v = \boldsymbol{b}_1 - \boldsymbol{b}_2$, Horn's method uses $\boldsymbol{a}_v = \boldsymbol{a}_1 - \boldsymbol{a}_c$ and $\boldsymbol{b}_v = \boldsymbol{b}_1 - \boldsymbol{b}_c$, where $\boldsymbol{a}_c$ and $\boldsymbol{b}_c$ are the centroids of $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ respectively.

Further, Eq. 6.5 is similar to the one used by Faugeras *et. al.* [60]. However, Faugeras *et. al.* [60]did not consider uncertainties in sensor measurements while esti-

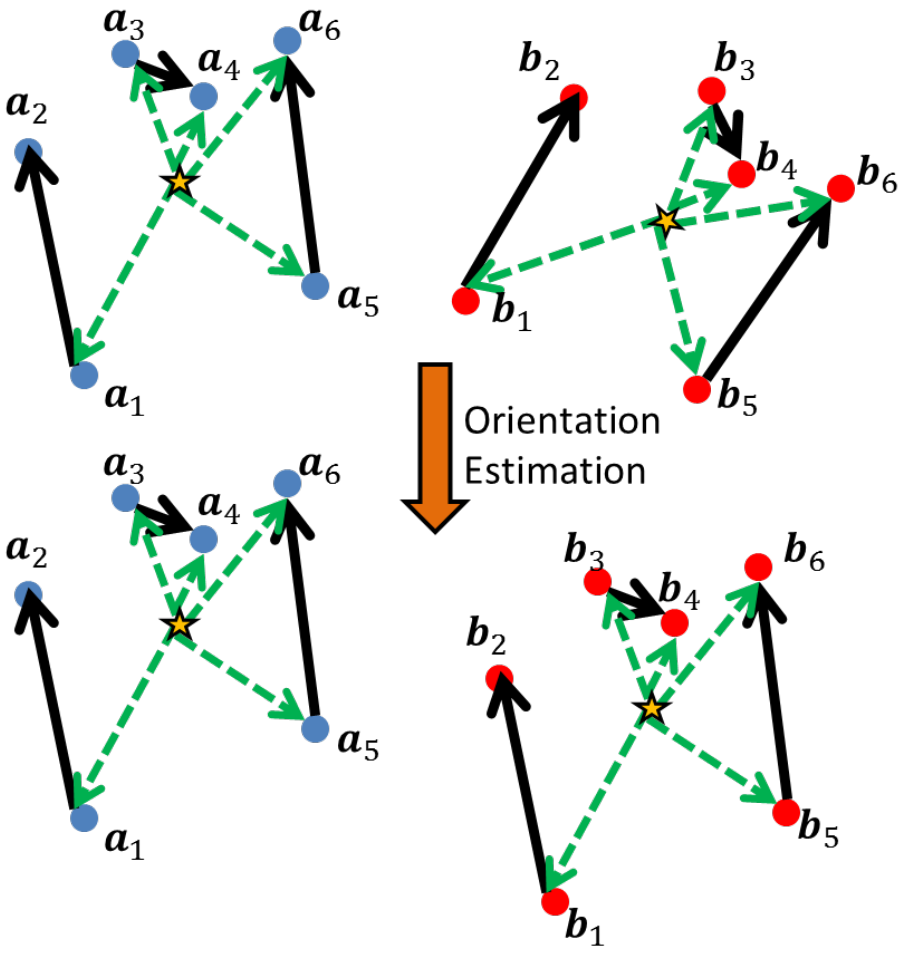


Figure 6-2: Blue points (left) indicate $\boldsymbol{a}_i$ and red points (right) indicate $\boldsymbol{b}_i$. Our approach constructs vectors $\boldsymbol{a}_v^{ij} = (\boldsymbol{a}_i - \boldsymbol{a}_j)$ and $\boldsymbol{b}_v^{ij} = (\boldsymbol{b}_i - \boldsymbol{b}_j)$ as shown by black arrows. The Bingham filter estimates the orientation between the black vectors. A standard implementation of Horn [4] on the other hand, finds the orientation between the green-dashed vectors. While the green-dashed vectors can only be constructed using a batch of measurements, the black vectors can be constructed from sequential measurements. The black vectors can be considered to be a special case of green-dashed vectors where one pair of data points are considered at a time with one of the points serving as the centroid.

mating $\widetilde{\boldsymbol{q}}$. In this work, we model the uncertainty in the sensor measurements $\boldsymbol{a}_i, \boldsymbol{b}_i$ using Gaussian distribution. Let $\boldsymbol{a}_i^s = \boldsymbol{a}_i + \delta\boldsymbol{a}_i$ and $\boldsymbol{b}_i^s = \boldsymbol{b}_i + \delta\boldsymbol{b}_i$, where $(\cdot)^s$ is a sensor measurement, and $\delta(\cdot)$ is the noise as sampled from a zero mean Gaussian,

$\mathcal{N}(0, \boldsymbol{\Sigma}^{(\cdot)})$. Eq. 6.5 can be rewritten as

$$\begin{aligned} \boldsymbol{H}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2)\widetilde{\boldsymbol{q}} &= \boldsymbol{0}, \\ \boldsymbol{H}(\boldsymbol{a}_1^s, \boldsymbol{a}_2^s, \boldsymbol{b}_1^s, \boldsymbol{b}_2^s)\widetilde{\boldsymbol{q}} + \boldsymbol{G}(\widetilde{\boldsymbol{q}})\boldsymbol{\mu} &= \boldsymbol{0}, \end{aligned} \tag{6.8}$$

where $\boldsymbol{\mu} = (\delta\boldsymbol{a}_1, \delta\boldsymbol{a}_2, \delta\boldsymbol{b}_1, \delta\boldsymbol{b}_2)^T$ and

$$\boldsymbol{G} = \begin{bmatrix} -\boldsymbol{W}_1 & \boldsymbol{W}_1 & \boldsymbol{W}_2 & -\boldsymbol{W}_2 \end{bmatrix}, \tag{6.9}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{4\times 3}$ are comprised of the last three columns of $\boldsymbol{F}_2(\widetilde{\boldsymbol{q}})$ and $\boldsymbol{F}_1(\widetilde{\boldsymbol{q}})$ respectively.

It can be shown that $\boldsymbol{G}(\widetilde{\boldsymbol{q}})\boldsymbol{\mu}$ is a zero mean Gaussian noise, $\mathcal{N}(0, \boldsymbol{Q})$, where the uncertainty $\boldsymbol{Q}$ is obtained analytically. To evaluate $\boldsymbol{Q}$, we make use of an important result from stochastic filtering theory which is described in Proposition 1 (pp. 90–91 of Jazwinski [146] and Appendix A of Chaukroun *et. al.* [143]). For the sake of completeness, we prove the proposition in the Appendix .1. We also provide a couple of examples that illustrate this proposition.

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

$$\boldsymbol{Q} = \boldsymbol{G}(\widetilde{\boldsymbol{q}})\boldsymbol{\Sigma}^{\mu}\boldsymbol{G}(\widetilde{\boldsymbol{q}})^T. \tag{6.10}$$

Linear Filter

In order to obtain an estimate of $\widetilde{\boldsymbol{q}}$ from Eq. 6.5, we use a Bingham distribution to model the uncertainty in $\widetilde{\boldsymbol{q}}$,

$$p(\widetilde{\boldsymbol{q}}) = \frac{1}{N_1}\exp(\widetilde{\boldsymbol{q}}^T \underbrace{\boldsymbol{M}_{k-1}\boldsymbol{Z}_{k-1}\boldsymbol{M}_{k-1}^T}_{\boldsymbol{D}_1}\widetilde{\boldsymbol{q}}). \tag{6.11}$$

If the pose was changing with time, then a suitable process model can be employed as shown in [54]. For now we restrict the analysis to static pose estimation and hence

do not consider a process model to evolve the pose estimate over time. Position measurements are obtained, which are in turn used to update the pose estimate. The pose is updated once for every pair of measurements received. The following is the probability of obtaining a sensor measurement $\boldsymbol{z}_k$, given the state $\widetilde{\boldsymbol{q}}_k$,

$$p(\boldsymbol{z}_k|\widetilde{\boldsymbol{q}}_k) = \frac{1}{N_2}\exp\left(-\frac{1}{2}(\boldsymbol{z}_k - \boldsymbol{h}(\widetilde{\boldsymbol{q}}_k))^T\boldsymbol{Q}_k^{-1}(\boldsymbol{z}_k - \boldsymbol{h}(\widetilde{\boldsymbol{q}}_k))\right), \tag{6.12}$$

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

From Eq. 6.8, we set the measurement to $\boldsymbol{z}_k = \boldsymbol{0}$ and measurement model to $\boldsymbol{h}(\widetilde{\boldsymbol{q}}_k) = \boldsymbol{H}\widetilde{\boldsymbol{q}}_k$. In a Bayesian update, the state $\widetilde{\boldsymbol{q}}_k$ is updated such that the predicted measurement $\boldsymbol{H}\widetilde{\boldsymbol{q}}_k$ is close to the actual measurement $\boldsymbol{z}_k$. Setting $\boldsymbol{z}_k = 0$ ensures that the state $\widetilde{\boldsymbol{q}}_k$ is updated so that $\boldsymbol{H}\widetilde{\boldsymbol{q}}_k = \boldsymbol{z}_k = \boldsymbol{0}$ (as required from Eq. 6.8).

Since $\boldsymbol{z}_k = \boldsymbol{0}$ is not a true measurement, it is often referred to as pseudo-measurement in literature ([187]). For a detailed discussion on deriving update equations using a linear pseudo-measurement model, refer to [188, 186]. Eq. 6.12 can be rewritten for our case as,

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

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

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

$$\begin{aligned} p(\widetilde{\boldsymbol{q}}_k|\boldsymbol{z}_k) &\propto p(\widetilde{\boldsymbol{q}}_k)p(\boldsymbol{z}_k|\widetilde{\boldsymbol{q}}_k) \\ &\propto \frac{1}{N_1}\exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{D}_1\widetilde{\boldsymbol{q}}_k\right)\frac{1}{N_2}\exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{D}_2\widetilde{\boldsymbol{q}}_k\right) \quad (6.13) \\ &\propto \exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{M}_k\boldsymbol{Z}_k\boldsymbol{M}_k^T\widetilde{\boldsymbol{q}}_k\right). \quad (6.14) \end{aligned}$$

And thus it can be seen that the maximum a posteriori estimate, $p(\widetilde{\boldsymbol{q}}_k|\boldsymbol{z}_k)$, is a Bingham distribution, where $\boldsymbol{M}_k\boldsymbol{Z}_k\boldsymbol{M}_k^T$ is obtained from the product of Binghams as shown in Eq. 2.24. As mentioned in Sec. 2.5, the mode of the distribution $\widetilde{\boldsymbol{q}}_k$, is the first column of $\boldsymbol{M}_k$.

It is worth noting that when no prior uncertainty information is available, $\boldsymbol{Q}_k$ can be obtained from Eq. 6.10. However, it can be shown that the rank of $\boldsymbol{Q}_k$ (obtained from Eq. 6.10), is at most three. But Eq. 6.12 requires $\boldsymbol{Q}_k$ to be invertible. Thus we assume prior uncertainty of $\widetilde{\boldsymbol{q}}$ to be very large in such cases which would allow us to use Proposition 1 and obtain an invertible $\boldsymbol{Q}$. The first few state updates would not produce meaningful results due to high uncertainty in $\widetilde{\boldsymbol{q}}$. But each update decreases the uncertainty and after a few updates, the filter starts producing meaningful results.

After updating $\widetilde{\boldsymbol{q}}_k$, we estimate $\boldsymbol{t}_k$ from Eq. 6.7. The prior and likelihood of $\boldsymbol{t}$ are

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

where the derivation for the likelihood and the definition of $\boldsymbol{R}_k$, $\boldsymbol{a}_c$, $\boldsymbol{b}_c$ are provided in Appendix .2.

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

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

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

$$\boldsymbol{t}_k = \left(\left(\boldsymbol{\Sigma}^{\boldsymbol{t}}_{k-1}\right)^{-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}\boldsymbol{W}_1\right)^{-1}\left(\left(\boldsymbol{\Sigma}^{\boldsymbol{t}}_{k-1}\right)^{-1}\boldsymbol{t}_{k-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}(\boldsymbol{W}_1\boldsymbol{a}_c - \boldsymbol{W}_2\boldsymbol{b}_c)\right) \tag{6.15}$$

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

$$\boldsymbol{\Sigma}^{\boldsymbol{t}}_k = \left(\left(\boldsymbol{\Sigma}^{\boldsymbol{t}}_{k-1}\right)^{-1} + \boldsymbol{W}_1^T\boldsymbol{R}_k^{-1}\boldsymbol{W}_1\right)^{-1}. \tag{6.16}$$

The above update equations are identical to Kalman filtering update [51]. Hence, the state is updated once for every pair of measurements received, until a convergence condition is reached, or maximum number of updates is reached.

Simultaneous Multi-measurement Update

So far we have considered only the case where the state is updated once per pair of measurements. However, such an approach can be inefficient when applied to pose estimation from stereo cameras or Kinect$^{\text{TM}}$. In such applications, one typically obtains several position measurements at each time instant and processing the measurements in a pairwise manner can be time consuming. In order to address this situation, we can rewrite Eq. 6.5 as:

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

$\boldsymbol{H}_j$ has the form as shown in Eq. 6.6, where $\boldsymbol{a}_v, \boldsymbol{b}_v$ are obtained from point-pairs constructed by subtracting random pairs of points or subtracting each point from the centroid (similar to [4]). Since the measurements are assumed to be independent, we have

$$\begin{aligned} p(\boldsymbol{z}_k|\widetilde{\boldsymbol{q}}_k) &= \prod_{j=1}^{m} \frac{1}{N_2^j} \exp\left(\frac{-1}{2}(\boldsymbol{H}_j\widetilde{\boldsymbol{q}}_k)^T\widetilde{\boldsymbol{q}}_k^{-1}(\boldsymbol{H}_j\widetilde{\boldsymbol{q}}_k)\right), \\ &= \frac{1}{N_3}\exp(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{D}_3\widetilde{\boldsymbol{q}}_k), \end{aligned} \tag{6.17}$$

where $\boldsymbol{D}_3 = \frac{1}{2}\sum_j \left(-\boldsymbol{H}_j^T\widetilde{\boldsymbol{q}}_k^{-1}\boldsymbol{H}_j\right)$ and $N_3 = \prod_{j=1}^{m} N_2^j$. Eq. 6.13 can be rewritten as

$$\begin{aligned} p(\widetilde{\boldsymbol{q}}_k|\boldsymbol{z}_k) &\propto \frac{1}{N_1}\exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{D}_1\widetilde{\boldsymbol{q}}_k\right)\frac{1}{N_3}\exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{D}_3\widetilde{\boldsymbol{q}}_k\right) \\ &\propto \exp\left(\widetilde{\boldsymbol{q}}_k^T\boldsymbol{M}_k\boldsymbol{Z}_k\boldsymbol{M}_k^T\widetilde{\boldsymbol{q}}_k\right), \end{aligned} \tag{6.18}$$

where $\boldsymbol{M}_k\boldsymbol{Z}_k\boldsymbol{M}_k^T$ is obtained from Bingham multiplication. $\widetilde{\boldsymbol{q}}_k$ and $\boldsymbol{t}_k$ are obtained as shown in Sec. 6.2.1.

6.2.2 Surface-normal Measurements

In some applications, in addition to position measurements, surface-normal measurements may also be available ([23, 113]). The following equation relates the surface-normals in the two frames,

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

where $\boldsymbol{n}_i^a$ are surface-normals in CAD-model frame and $\boldsymbol{n}_i^b$ are surface-normals in the sensor frame. Similar to the derivation in the case of position measurements (see

Eq. 6.17), we obtain,

$$p(\boldsymbol{z}_k|\widetilde{\boldsymbol{q}}_k) = \frac{1}{N_4}\exp(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_4 \widetilde{\boldsymbol{q}}_k), \tag{6.19}$$

where $\boldsymbol{D}_4 = \frac{1}{2}\sum_i \left(-\boldsymbol{J}_i^T \boldsymbol{S}_k^{-1} \boldsymbol{J}_i\right) + \frac{1}{2}\sum_j \left(-\boldsymbol{H}_j^T \widetilde{\boldsymbol{q}}^{-1} \boldsymbol{H}_j\right)$, $\boldsymbol{S}_k$ is the pseudo-measurement uncertainty. Thus, we have

$$\begin{aligned} p(\widetilde{\boldsymbol{q}}_k|\boldsymbol{z}_k) &\propto \frac{1}{N_1}\exp(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_1 \widetilde{\boldsymbol{q}}_k)\frac{1}{N_4}\exp(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{D}_4 \widetilde{\boldsymbol{q}}_k), \\ &\propto \exp(\widetilde{\boldsymbol{q}}_k^T \boldsymbol{M}_k \boldsymbol{Z}_k \boldsymbol{M}_k^T \widetilde{\boldsymbol{q}}_k). \end{aligned}$$

Instead of using a Gaussian distribution to model the uncertainty in the surface-normal measurements, a von Mises Fisher distribution or a Kent distribution may be chosen (depending on if the uncertainty is isotropic or anisotropic), as shown by Billings *et. al.* [23]. For calculating the uncertainties using Proposition 1, the covariance of the von Mises and Kent distributions can be calculated as shown by Hillen *et. al.* [189] and Balov *et. al.* [190] respectively.

6.2.3 Pose Measurements

Systems that use pose measurements for model update typically have the following general form ([149, 33])

$$\boldsymbol{A}\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B} = \boldsymbol{0}, \tag{6.20}$$

where $\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{B} \in SE(3)$. These problems are generally referred to as 'hand-eye calibration'. $\boldsymbol{A}$ and $\boldsymbol{B}$ are pose-measurements and $\boldsymbol{X}$ is the desired transformation to be estimated. While variants of this problem exist in the form of $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{B}\boldsymbol{Y}$ (also known as hand-eye robot-world calibration) ([158, 191]), these problems can also be reduced to the form of $\boldsymbol{A}\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}$ by using relative measurements ([192]). Let $\widetilde{\boldsymbol{a}}^r, \widetilde{\boldsymbol{b}}^r \in \mathbb{R}^4$ be the unit quaternion and $\boldsymbol{a}^t, \boldsymbol{b}^t \in \mathbb{R}^3$ be the translation, parameterizing

$\boldsymbol{A}$ and $\boldsymbol{B}$ respectively. For Eq. 6.20, we have

$$\widetilde{\boldsymbol{a}}^r \odot \widetilde{\boldsymbol{q}} - \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}^r = \mathbf{0}, \quad \text{and,} \tag{6.21}$$

$$\widetilde{\boldsymbol{a}}^t \odot \widetilde{\boldsymbol{a}}^r \odot \widetilde{\boldsymbol{q}} + \widetilde{\boldsymbol{a}}^r \odot \widetilde{\boldsymbol{t}} \odot \widetilde{\boldsymbol{q}} - \widetilde{\boldsymbol{t}} \odot \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}^r - \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}^t \odot \widetilde{\boldsymbol{b}}^r = \mathbf{0}, \tag{6.22}$$

where $\widetilde{\boldsymbol{a}}^t \triangleq (0, (\boldsymbol{a}^t)^T)^T$, $\widetilde{\boldsymbol{b}}^t \triangleq (0, (\boldsymbol{b}^t)^T)^T$. From Eq. 6.21 and Eq. 2.11, we obtain

$$\boldsymbol{L}^r(\widetilde{\boldsymbol{a}}^r, \widetilde{\boldsymbol{b}}^r)\widetilde{\boldsymbol{q}} = \mathbf{0}, \quad \text{where,} \tag{6.23}$$
$$\boldsymbol{L}^r(\widetilde{\boldsymbol{a}}^r, \widetilde{\boldsymbol{b}}^r) = \boldsymbol{F}_1(\widetilde{\boldsymbol{a}}^r) - \boldsymbol{F}_2(\widetilde{\boldsymbol{b}}^r) \in \mathbb{R}^{4\times 4}.$$

Eq. 6.23 is linear in $\widetilde{\boldsymbol{q}}$, as was the case in the previous sections. Thus we follow a similar analyses to estimate $\widetilde{\boldsymbol{q}}_k$ and the associated uncertainty.

Once we obtain $\widetilde{\boldsymbol{q}}_k$, its value is substituted in Eq. 6.22 and the terms are rearranged to obtain

$$\boldsymbol{L}^t(\widetilde{\boldsymbol{a}}^r, \boldsymbol{a}^t, \widetilde{\boldsymbol{b}}^r, \boldsymbol{b}^t)\widetilde{\boldsymbol{t}} + \widetilde{\boldsymbol{\tau}}_3 = \mathbf{0}, \quad \text{where,} \tag{6.24}$$
$$\boldsymbol{L}^t = \boldsymbol{F}_1(\widetilde{\boldsymbol{a}}^r) - \boldsymbol{F}_2(\widetilde{\boldsymbol{q}}_k \odot \widetilde{\boldsymbol{b}}^r \odot \widetilde{\boldsymbol{q}}_k^*),$$
$$\widetilde{\boldsymbol{\tau}}_3 = \widetilde{\boldsymbol{a}}^t \odot \widetilde{\boldsymbol{a}}^r - \widetilde{\boldsymbol{q}}_k \odot \widetilde{\boldsymbol{b}}^t \odot \widetilde{\boldsymbol{b}}^r \odot \widetilde{\boldsymbol{q}}_k^*.$$

Eq. 6.24 can be simplified as shown

$$\overline{\boldsymbol{L}^t}\boldsymbol{t} + \boldsymbol{\tau}_3 = \mathbf{0},$$

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

We use a maximum a priori estimate to obtain $\boldsymbol{t}_k$ and $\boldsymbol{\Sigma}_k^{\boldsymbol{t}}$ similar to Eq. 6.15 and Eq. 6.16,

$$\boldsymbol{t}_k = \left(\left(\boldsymbol{\Sigma}_{k-1}^{\boldsymbol{t}}\right)^{-1} + \overline{\boldsymbol{L}^t}^T \boldsymbol{R}_k^{-1} \overline{\boldsymbol{L}^t}\right)^{-1} \left(\left(\boldsymbol{\Sigma}_{k-1}^{\boldsymbol{t}}\right)^{-1} \boldsymbol{t}_{k-1} - (\boldsymbol{L}^t)^T \boldsymbol{R}_k^{-1} \boldsymbol{\tau}_3\right), \tag{6.25}$$

$$\boldsymbol{\Sigma}_k^{\boldsymbol{t}} = \left(\left(\boldsymbol{\Sigma}_{k-1}^{\boldsymbol{t}}\right)^{-1} + \overline{\boldsymbol{L}^t}^T \boldsymbol{R}_k^{-1} \overline{\boldsymbol{L}^t}\right)^{-1}, \tag{6.26}$$

where the uncertainty $\boldsymbol{R}_k$ can be obtained following steps similar to those shown in Sec. 6.2.1. See Appendix .2 for the equation of $\boldsymbol{R}_k$.

6.2.4 Dealing with Unknown Data-association

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

Let ψ be the model shape that is often available in the form of a triangulated mesh. Let $\boldsymbol{b}_i^s \in \mathbb{R}^3$ be the sensor measurements in the sensor's reference frame. If we knew the point on ψ that is associated with $\boldsymbol{b}_i^s$, then we could follow the approach as shown in Sec. 6.2.1. However, we do not know this data association, and so we find the point $\boldsymbol{a}_i \in \psi$, such that

$$\boldsymbol{a}_i = \underset{\boldsymbol{a}\in\psi}{\operatorname{argmin}} \, |\boldsymbol{a} - Rot(\widetilde{\boldsymbol{q}}_k)\boldsymbol{b}_i^s - \boldsymbol{t}_k|. \tag{6.27}$$

This approach is referred to as 'closest point' correspondence and is often used in methods such as ICP [11]. An alternate approach involves using a probabilistic criteria ('most likely' correspondence developed by Billings *et. al.* [23]) instead of Eq. 6.27,

$$\boldsymbol{a}_i = \underset{\boldsymbol{a}\in\psi}{\operatorname{argmax}} \frac{1}{N} \exp\left(-\frac{1}{2}\boldsymbol{v}_k^T \boldsymbol{S}_k \boldsymbol{v}_k\right), \tag{6.28}$$

$$= \underset{\boldsymbol{a}\in\psi}{\operatorname{argmin}} \, \boldsymbol{v}_k^T \boldsymbol{S}_k \boldsymbol{v}_k, \tag{6.29}$$

where $\boldsymbol{v}_k = \boldsymbol{a} - Rot(\widetilde{\boldsymbol{q}}_k)\boldsymbol{b}_i^s - \boldsymbol{t}_k$, and $\boldsymbol{S}_k = Rot(\widetilde{\boldsymbol{q}}_k)\boldsymbol{\Sigma}^{\boldsymbol{b}_i^s} Rot(\widetilde{\boldsymbol{q}}_k)^T + \boldsymbol{\Sigma}^{\boldsymbol{a}}$. Eq. 6.28 does not take into account the uncertainty associated with $\widetilde{\boldsymbol{q}}_k$ and $\boldsymbol{t}_k$. While it is straightforward to estimate the uncertainty in the pose and incorporate in Eq. 6.28, we suspect that doing so can result in worse data associations, especially when the initial pose-uncertainty is very high. Additionally, both the 'closest point' as well as 'most likely' paradigms are equivalent when the uncertainties $\boldsymbol{\Sigma}^{\boldsymbol{b}_i}, \boldsymbol{\Sigma}^{\boldsymbol{a}}$ are isotropic (see Billings *et. al.* [23]).

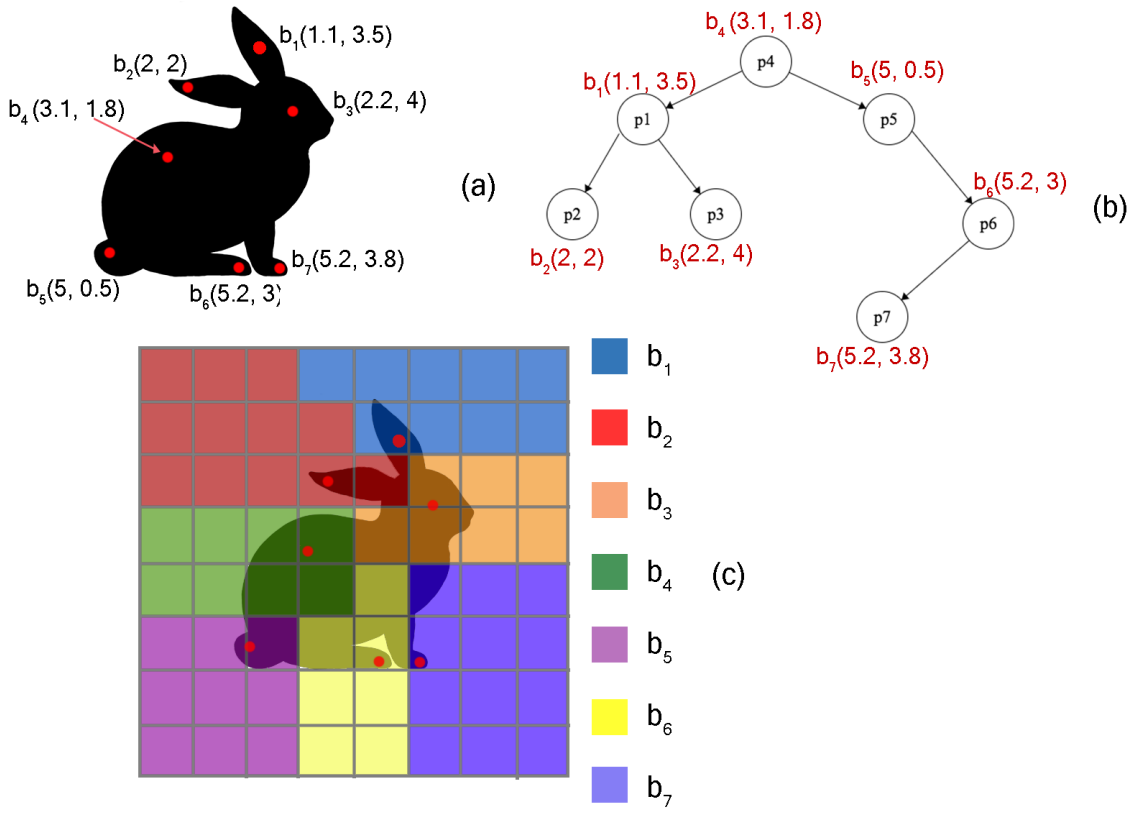


Figure 6-3: (a) Shows a representative point cloud of a model. For ease of demonstration, we show a 2D case. Let the model have 7 points as shown by red dots. (b) k-d tree constructed for the model points. (c) Look-up table approach is illustrated. A grid is created around the model. The closest point from each grid center to the model is computed using k-d tree. Each grid-color shows an index corresponding to the closest point on the model. Given a new sensor point, we first transform it to the model frame (using the best estimate of transformation). We then find which grid it belongs to and return the precomputed closest point.

k-d Tree Search:

A naive implementation for optimizing Eq. 6.27 or Eq. 6.28 would require checking every point in ψ and find the one that minimizes the objective. However, such an approach takes $O(n)$ time on an average, which can be practically infeasible in many situations. As a result a popular approach to search for the optimal point in the model involves using a k-d tree originally developed by Friedman *et. al.* [193] (see Fig. 6-3(b)). A k-d tree with n points takes $O(\log n)$ time on an average per search. There are different ways to build k-d trees depending on which coordinate is chosen

for splitting the data. One often chooses the coordinate with the largest spread. Building a k-d tree with n points has a complexity of $O(n \log n)$, where amedian of medians algorithm is used to select the median at each level of the tree [194]. One variant is the dyadic tree which cycles through the coordinates and splits the data at the midpoint.There are several other variants of the k-d tree such as ball tree [195], Vantage-point tree [196], etc.

Principal Direction Tree Search:

One may also use a principal direction (PD) tree originally developed by William *et. al.* [197], though it was not called a PD-tree then. Others such as Verma *et. al.* [198] used the term PD-tree. The primary difference between the k-d and PD-tree is that each node of the PD-tree has a local coordinate system that is oriented based on the spread of the points in ψ instead of being axis-aligned with the model's reference frame. Such a tree can result in more compact geometric bounds of nodes within the PD-tree and hence provide a boost to the search efficiency. Billings *et. al.* [23] use a PD-tree search in their algorithm and demonstrate improved performance over k-d tree search. A variant of this method is the random projection tree (RP tree) developed by Dasgupta *et. al.* [199]. In this tree a random vector is chosen and the tree is split along the median of the data projected onto this vector. The RP tree is computationally faster than PD-tree but produces lower accuracy results. McCartin *et. al.* [200] have developed an approximate PD tree (APD tree) that has the computational complexity of an RP tree with the accuracy comparable to a PD-tree.

Compared with k-d trees, PD trees and RP trees yield better space partitions and thus lead to better order for visiting the points because the partition hyperplanes are less limited and more flexible than those in k-d trees. However, in the query stage, the time overhead in PD trees and RP trees is larger because the branching step, determining which child of an internal node is next visited, requires an inner-product operation that consists of O(d) multiplications and O(d) additions while it costs only O(1) in k-d trees. Therefore, in high-dimensional problems k-d trees usually achieve better accuracy than PD trees within the same search time, as observed by

Wang *et. al.* [201].

Fast Look-up Table Search:

In this work, we have developed a faster (albeit more approximate) approach to search for the closest point to the model. We refer to this approach as 'look-up table' approach. As shown in Fig. 6-3(c), we use a uniform grid to discretize the space around the model. We then compute the closest point from the center of each grid to the model using a k-d tree. The index of the closest point on the model to each grid point $\boldsymbol{g}_i$ is stored in a look-up table. Such a look-up table needs to be constructed once before the start of the experiments. When we obtain a sensor measurement $\boldsymbol{b}_i$, we find the closest grid-center to $(Rot(\widetilde{\boldsymbol{q}}_k)\boldsymbol{b}_i + \boldsymbol{t}_k)$, and return the corresponding pre-computed closest point. Since the grid is uniform, the computation of the closest grid center can be done in $O(1)$.

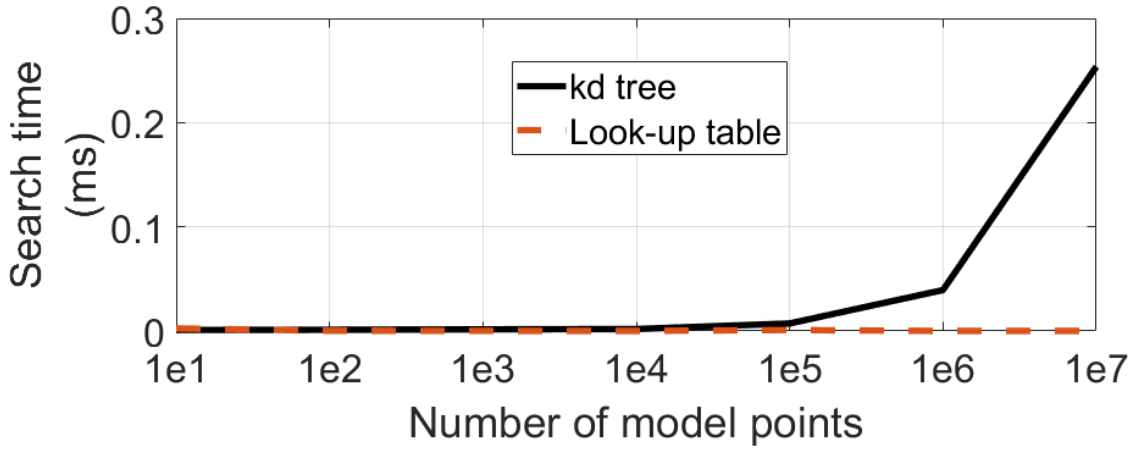


Figure 6-4: Plot shows the time taken per search vs number of model points for k-d tree and look up table approach. Irrespective of the number of model points, the search time with look-up table approach remains of the same order ($\approx 2 \times 10^{-3}$ ms). k-d tree, on the other hand, takes 10^{-1} ms for 10 model points (which is 2 times faster than look-up table) and 0.25 ms for 1 million model points (which is 100 times slower than look-up table).

We choose a uniform grid in a box, which is three times the largest dimension of the model, with 100 grid points in each dimension. The center of the grid space is chosen to coincide with the center of the model. An advantage of this search method is that

the closest point computation is carried offline, and finding the closest grid-center is an inexpensive operation. Fig. 6-4 shows that when the model has $> 10,000$ points, the look-up table approach outperforms k-d tree by several orders of magnitude[1]. However, a shortcoming of this approach is that the density of the grid dictates the accuracy of the result, because every point inside each grid is assigned the same closest point. Increasing the density of the grid would result in more accurate results at the cost of increase memory storage to save the lookup table. It is also worth noting that increasing the grid size would increase the memory requirements as well. For example the look-up table for a grid with 100 grid points in each dimension has a size of 8.91 kB in Matlab.

It is interesting to note that Fig. 6-3 is actually a grid approximation of the Voronoi diagram of the model points. The grid approximation allows for quick retrieval of closest point, as opposed to saving the Voronoi cells and checking for which Voronoi cell a given point belongs to. If we were to consider the model as a continuous surface instead of a set of points, the look-up table approach can be modified to store the closest point on the model surface, instead of storing the index of the model point. This could increase the storage space requirements as we would be storing three floats for each grid element as opposed to an integer.

It is worth noting that prior work such as 2D image registration of Paragios *et. al.* [202], parametric surface fitting of Flory *et. al.* [203], and KinectFusion of Izadi *et. al.* [15] compute a signed Euclidean distance between each grid point and the model, and use this information to minimize an energy function to bring the sensed points close to the model. Instead of finding the minimum Euclidean distance from grid point to the model, if we were to store the coordinates of the model point that results in the minimum distance, then those approaches would revert to our look-up table approach.

[1]The time reported is for a script written in MATLAB R2017b software from MathWorks, running on a ThinkPad T450s computer with 8 GB RAM and intel i7 processor.

6.3 Results

In this section, we consider two scenarios for using the Bingham distribution-based linear filter– 1) known data-association and 2) unknown data-association. Without loss of generality, we choose the following values for all experiments, $\boldsymbol{M}_0 = \boldsymbol{I}_{4\times4}$, $\boldsymbol{Z}_0 = diag(0,-1,-1,-1) \times 10^{-300}$ [2], which represents an uninformative prior with high initial uncertainty.

6.3.1 Known Data-association

Simulation Example– Point Cloud Registration:

In this section, we assume that the correspondence between points $\boldsymbol{a}_i \in \mathbb{R}^3$ and $\boldsymbol{b}_i \in \mathbb{R}^3$ are known, and estimate the pose between the frames that these two point sets lie in.

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

The linear BF is used to estimate the pose in each of the experiments in a sequential manner. This procedure is repeated 1000 times with different datasets and different transformations that are randomly generated. The results are compared with DQF (from Chapter 4), Moghari-UKF [5] and Pennec-EKF [6]. We also compare the results with an incremental variant of Horn's method (Horn-Inc) [4]. In this method we find estimate the pose once for every 4 measurements obtained using the estimate from the previous update as an initial condition.

[2] 10^{-300} is the smallest positive normalized floating-point number in IEEE® double precision.

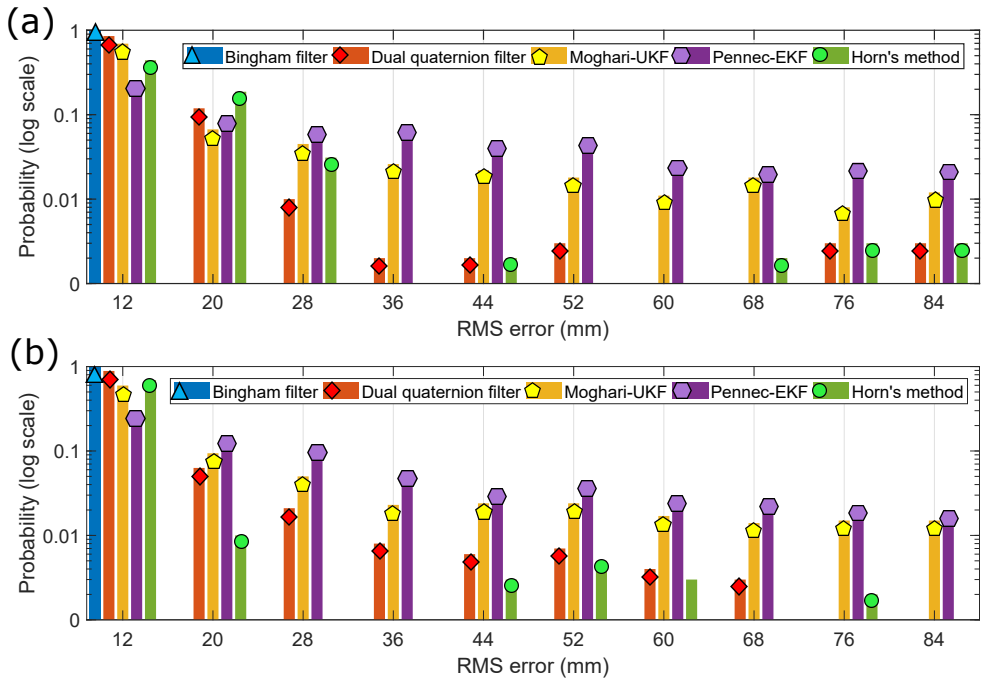


Figure 6-5: Histogram shows the RMS errors for the Bingham filter (BF), dual quaternion filter (DQF) from Chapter 4, unscented Kalman filter of [5] (Moghari-UKF), extended Kalman filter of [6] (EKF) and an incremental variant of Horn's method ([4]). (a) The results shown are for Expt. 3, where the sensed points have a noise uniformly drawn from [-10 mm 10 mm]. The BF is most accurate with an average RMS error of 10.30 mm and a 100% success rate. (b) The results shown are for Expt. 5, where the sensed points have a noise obtained from a Gaussian distribution $\mathcal{N}(\mathbf{0}, diag(10, 10, 10))$. The BF is the most accurate with an average RMS error of 4.95mm and a 100% success rate.

Table 6.1: Mean RMS errors for experiments involving three different levels of uniform measurement noise.

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

For each of the methods, if the RMS error at the end of any experiment is greater than 250 mm (the size of the workspace considered), then we consider the experiment to have failed. The average RMS errors over the first three experiments along with the percentage of successful runs are tabulated in Table 6.1. Note that we do not include failed experiments in our computation of RMS errors.

Fig. 6-5(a) shows the histogram of errors for Expt. 3. The BF always estimates the pose with the lowest RMS error. The RMS error of DQF and Moghari-UKF are both small, but larger than the BF. Due to the large initial orientation chosen, they get trapped in local minima sometimes, which is captured by the bars at higher RMS errors in Fig. 6-5(a). While Horn-Inc performs better than Moghari-UKF when the noise is small, higher noise results in the Horn-Inc performing worse. Pennec-EKF performed the worst in all the three experiments. For Expt. 3, the average run time for the BF is 26 ms, compared to 9 ms of DQF, 3.8 ms of Horn-Inc, 130 ms of Moghari-UKF and 17 ms of Pennec-EKF.

The average RMS errors for Experiment 4 and Experiment 5 along with the percentage of successful runs are tabulated in Table 6.2.

Table 6.2: Mean RMS errors for experiments involving two different levels of Gaussian noise in the measurements.

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

Fig. 6-5(b) shows the histogram of errors for Expt. 3. While the trend is similar to the case of the first three experiments, all the approaches perform better than in the case of uniform noise. This is an expected behavior, since the filters are developed to handle Gaussian noise in the measurements. BF once again is most accurate among all the methods.

Real-world Example: Registering Camera and Robot Frame:

Fig. 6-6 shows an arm of a da Vinci® surgical robot (Intuitive Surgical Inc., Mountain View, CA) mounted on a table, and a stereo camera (ELP-1MP2CAM001 Dual Lens) mounted on a rigid stand. The relative pose between the robot's frame and the camera's frame is fixed, and needs to be estimated. To estimate this pose, the robot is telemanipulated in arbitrary paths and the location of tip of the robot $\boldsymbol{a}_i$ is computed in the camera frame by segmenting the tip from the stereo image and estimating its center. The position of the tip in the robot frame, $\boldsymbol{b}_i$ is obtained from the kinematics of the robot. The pose between the points $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ can be obtained as shown in Sec. 6.3.1. Table 6.3 shows the RMS error and the time taken for estimation by the

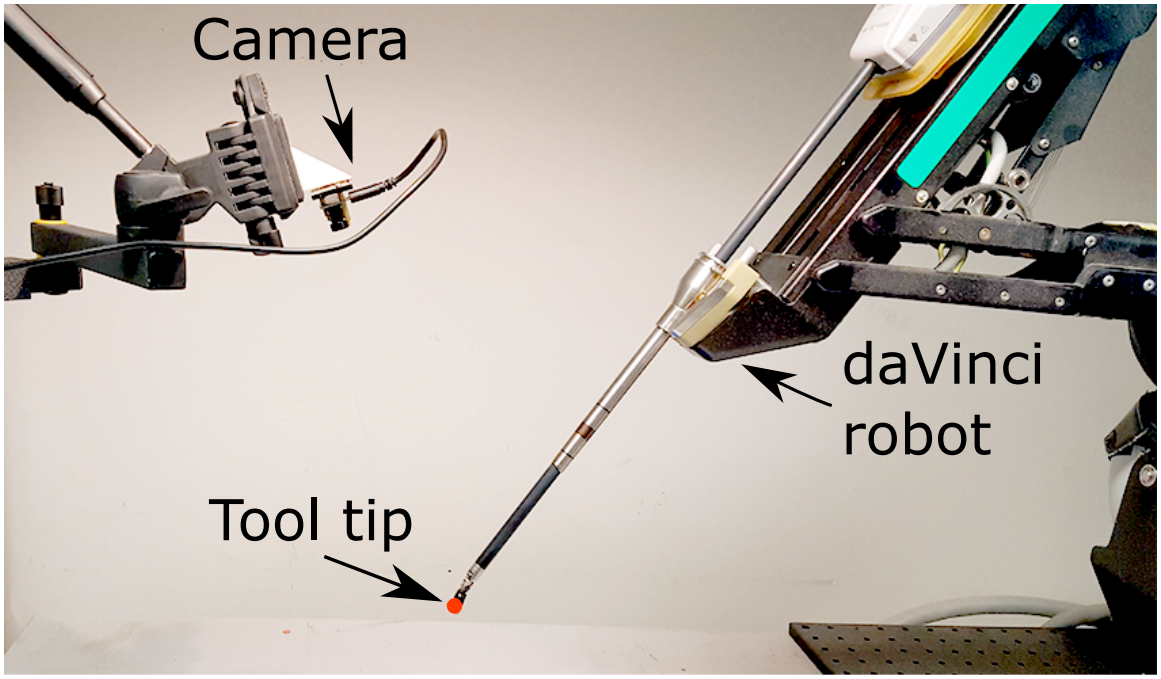


Figure 6-6: A spherical tool tip is attached to the daVinci robot. The tip is tracked using a stereo camera, which is held in a fixed position. As the robot is telemanipulated, the spherical tool-tip is tracked using the stereo camera, and the relative pose between the camera frame and the robot frame is estimated.

BF approach using pairwise updates, using 20 simultaneous measurement-pairs per update (abbreviated as BFM-20 in the Table), and an incremental Horn's method (Horn-Inc) where the pose is estimated once per 20 measurements. We evaluate the results for three termination criteria

1. Criteria 1: If the change in the translation and rotation fall below a set threshold, the algorithm is terminated. In this example we use 1mm and 0.1° as threshold for translation and rotation respectively.

2. Criteria 2: If the conditions of Criteria 1 are satisfied and the uncertainty in the rotation and translation fall below a set threshold. In this example we choose the threshold for uncertainty in translation as 10^{-4} for the largest eigenvalue of $\boldsymbol{\Sigma}_k^{\boldsymbol{t}}$ and -10^3 for the largest non-zero diagonal element of $\boldsymbol{Z}_k$.

3. Criteria 3: The algorithms are terminated after processing a set number of measurements. In this example we choose this number $n = 200$.

Table 6.3: Experimental results for robot-camera registration

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

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

When all three algorithms are run for 200 measurements, the accuracy of BFM-20 is the highest. Unlike BF and BFM-20, after processing 200 points, Horn-Inc has a higher RMS error. This is because in the filtering approaches, information from previous measurements are 'baked' into the current estimate by the uncertainty update; which is absent in Horn-Inc. Overall, Horn-Inc takes lowest computation

time and performs better than BF, but produces higher errors than BFM-20. We also perform a batch optimization using all 200 points and Horn's method (abbreviated as Horn-Batch in the table), as well as Bingham filter with a single update using 200 point measurements (Abbreviated as BFM-200). Both BFM-200 and Horn-Batch converge to the same RMS error and take comparable computation time, with the BFM-200 taking 0.14 ms longer due to additional computations involving uncertainty update.

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

Table 6.4: Simulation results for robot-camera registration

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	Time (ms)	RMS (mm)
Actual	11.79	261.49	27.29	178.25	7.29	-130.46	-	-
BFM-20	11.78	261.45	27.52	178.38	7.33	-130.35	42	2.79
Horn-Inc	20.67	259.21	27.52	177.08	5.73	-131.42	34	3.66

Since we did not observe many outliers in the experiment, we created a simulated example to test our approach in a more challenging case. In this simulation experiment, we take the measurements from the robot experiment described above and randomly chose 16% of the points to be outliers. We then estimate the pose using BFM-20 and Horn-Inc as shown in Fig. 6-7(b). In the absence of outlier detection both BFM-20 and Horn-Inc are highly inaccurate due to the presence of outliers. Upon using outlier detection, both the methods perform well, with the BFM-20 producing smoother estimates with lower RMS error than Horn-Inc.

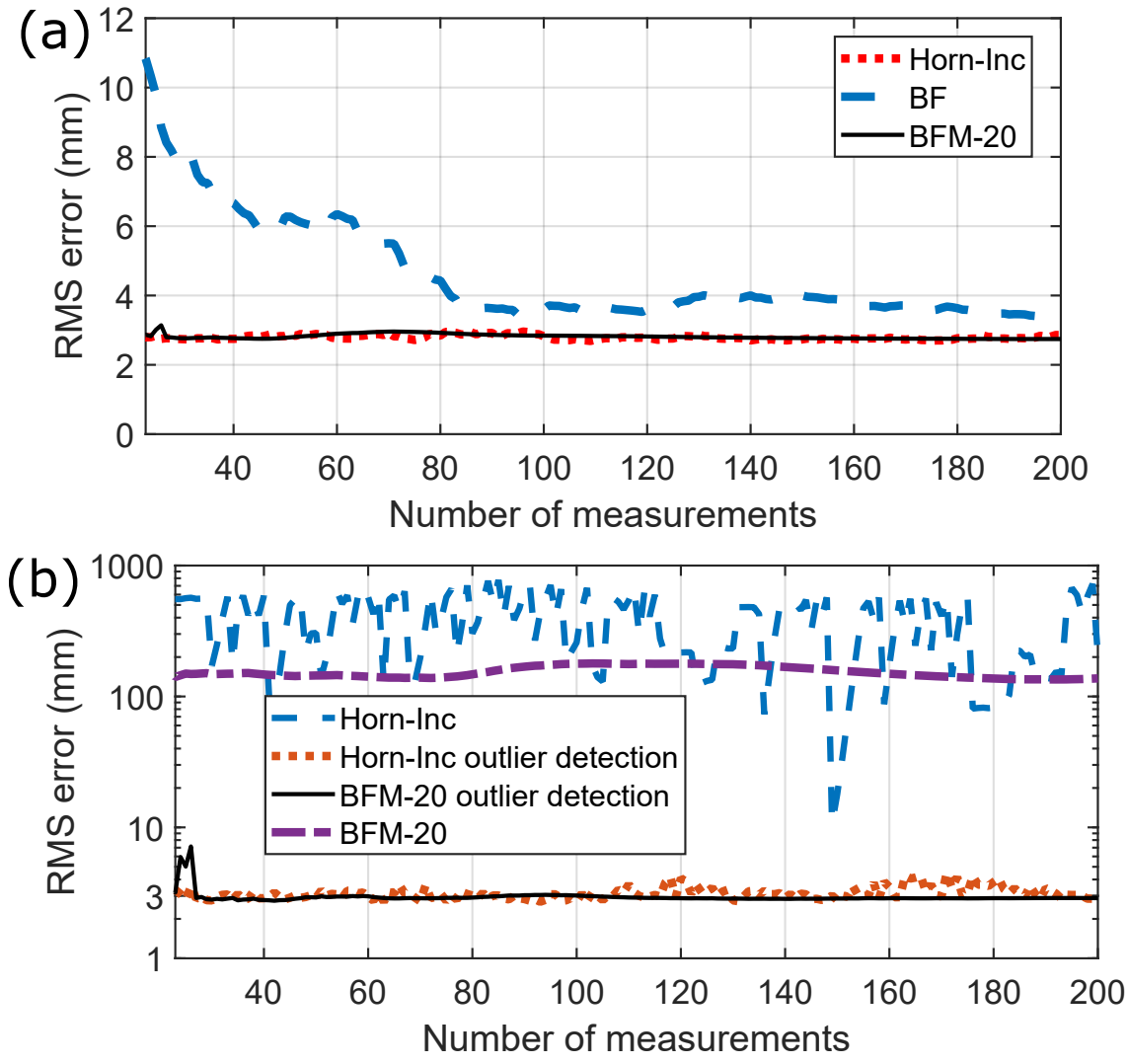


Figure 6-7: (a) Experimental results for robot-camera registration. BFM-20 produces the most accurate pose estimation, while Horn-Inc is the fastest and BF is the most erroneous. (b) Robot-camera registration results in the presence of 16% measurement outliers. In the absence of outlier detection both BFM-20 and Horn-Inc perform poorly. Outlier detection improves both BFM-20 and Horn-Inc, with the BFM-20 being smoother and more accurate.

6.3.2 Unknown Data-association

Simulation Example– Point-cloud Registration

In this section we assume that the points $\boldsymbol{a}_i$ and surface-normals $\boldsymbol{n}_i^{\boldsymbol{a}}$ are the vertices and normals respectively, of a triangulated mesh. Fig. 6-8 shows the triangulated mesh in the shape of a bunny [7], which has 86,632 triangles.

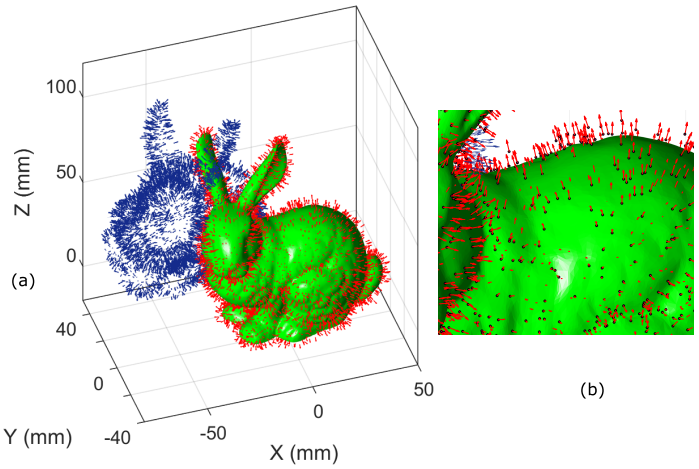


Figure 6-8: (a) Triangulated mesh of a Stanford bunny [7] is shown in green. Blue arrows represent initial location and red arrows represent estimated location of points and surface-normals. (b) Zoomed up view shows that the estimated location of points accurately rests on the triangulated mesh and the estimated direction of the surface-normals aligns well with the local surface normal. The Bingham filter takes 2.4s in MATLAB and 0.08s in C++ to estimate the pose.

We randomly pick 5000 points from the triangulated mesh and to each coordinate of the points, add a noise uniformly drawn from [-2 mm, 2 mm]. For each $(\boldsymbol{b}_i, \boldsymbol{n}_i^{\boldsymbol{b}})$, the correspondence is obtained by finding the closest point-normal pair $(\boldsymbol{a}_i, \boldsymbol{n}_i^{\boldsymbol{a}})$ on the triangulated mesh. We estimate the pose using the BF with 20 simultaneous multi-measurements as described in Sec. 6.2.1. Fig. 6-9 shows the RMS error vs number of simultaneous measurements used. Update based on one pair of measurements results in a local optimum (RMS error is $\approx$ 70 mm as shown in Fig. 6-9). However, the performance drastically improves when $>$ 10 simultaneous measurements are used.

The penultimate row of Table 6.5 shows the pose parameters as estimated by the BFM-20. We also estimate the pose using 20 simultaneous surface-normal and position measurements (abbreviated as BFN-20 in Table 6.5). The RMS error for the

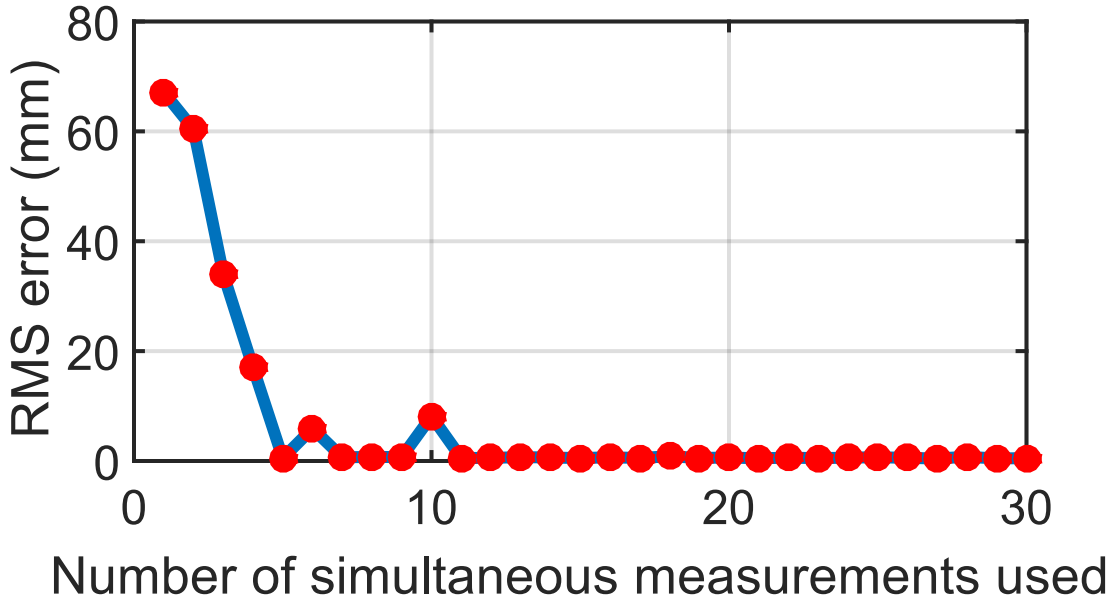


Figure 6-9: Plot shows RMS error upon convergence versus number of simultaneous measurements used. The more the number of simultaneous measurements used, the lower is the RMS error.

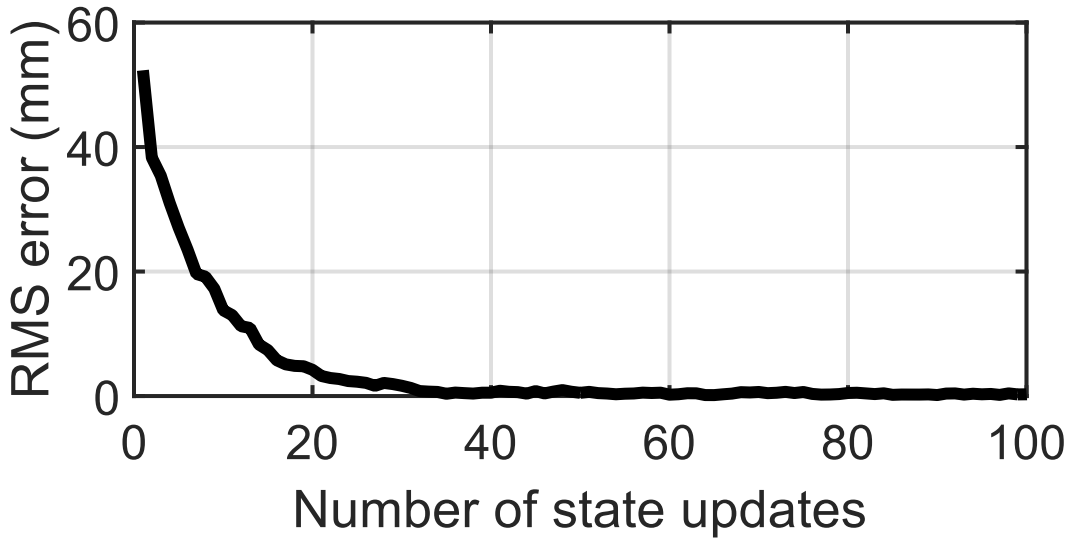


Figure 6-10: Plot shows the RMS error in the pose vs number of state updates as estimated by the Bingham filter using 20 simultaneous position and normal measurements in each update. The estimate converges in around 40 iterations.

BFN-20 is lower than the BFM-20, but the time taken is higher because of additional computations in the correspondence step involving surface-normals. Fig. 6-8(a) shows the initial position of the surface-normals and point locations with blue arrows and the

Table 6.5: Results for Registration of points and surface-normals to geometric model

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

BFN-20 estimated surface-normals and point locations with red arrow. The zoomed up image Fig. 6-8(b) shows that our approach accurately registers the points as well as aligns the surface-normals to the triangulated mesh.

Table 6.5 also shows the pose parameters as estimated by ICP [11], DQF, Pulli's method [28], Go-ICP [18] and IMLOP [25]. For the sake of a fair comparison, we use k-d tree search for the correspondence in all the methods except IMLOP. The correspondence criteria of IMLOP does not allow the use of a k-d tree and the authors use a specialized PD-tree search. Go-ICP takes 0.8s for estimating the pose parameters. However, we do not report the time taken for Go-ICP in Table 6.5 as the code originally supplied by the authors runs in C++ while all the other algorithms run on Matlab. BFM-20 implemented in Matlab takes less time than Go-ICP and is more accurate as well. This improvement in accuracy is attributed to the fact that Go-ICP does not consider uncertainties in measurements, while our approach does.

BFM-20 and BFN-20 are accurate and orders of magnitude faster than all the methods. Fig. 6-10 shows the RMS error at the end of each update step for BFN-20. The RMS error reduces to < 0.6mm at around 40 state updates. To obtain the same accuracy as DQF and ICP ($\approx$ 2mm), both BFM-20 and BFN-20 take $\approx$ 30 state updates, which takes 0.28s. The accuracy of Pulli is greater than ICP because it uses surface-normal information to prune the correspondence choices, which greatly helps with the registration. While Go-ICP does not use any surface-normal information, it performs a global search and hence produces results that are more accurate than ICP

and Pulli. IMLOP uses the point and surface-normal information and is as accurate as BFN-20 but takes several orders of magnitude more computation time. Furthermore, BFN-20 and BFM-20 produce serial updates and can be terminated using a criteria as described in Sec. 6.3.1. This is not possible in the case of IMLOP, which is a batch processing method and uses all the measurements to produce the pose estimate.

Simulation Example– Comparing k-d tree vs Look-up Table

In this section we evaluate the accuracy and the time taken for registering points to a model using k-d tree and look-up table-based approach. For this we consider the Lucy model from Stanford point cloud library dataset [7]. The model has 1.2 million triangle vertices. We first scale the model so that it fits within a cube of size 1 unit. We randomly sample 2500 points from this model and add a noise to it that is sampled from $\mathcal{N}(\mathbf{0}, diag(4,4,4)\times 10^{-4})$. We then apply a known transformation to these points and then try to estimate that transformation using the BFM with 20 simultaneous measurements. The transformation applied is $(0.01, -0.02, 0.05)$ units in translation and $(10, -10, 25)^\circ$ in rotation about each axis. We evaluate the performance of our approach using k-d tree as well as look-up table with varying grid sizes. The results are shown in Table 6.6. Note that the time reported does not include the time taken to construct the k-d tree or the look-up table [3]. As expected the k-d tree performs the best in terms of accuracy. The look-up table is several magnitudes of order faster. As we increase the grid density the accuracy of the look-up table increases as well, while search time remains constant.

Real-world Example– Point-cloud Stitching

Stereo imaging devices such as the Microsoft Kinect™ offer colored point cloud data (RGB-D: color and depth data), which is generated using a structured light based depth sensor. The Kinect™ is widely used in robot navigation [204] and object manipulation [9]. In this work, we align a pair of point cloud data obtained from

[3]The results are reported for a C++ implementation running on a MacBook Air, Processor 1.6 GHz Intel Core i5, Memory 8 GB 1600 MHz DDR3.

Table 6.6: RMS error and time taken to register points to a geometric model with 1.2 million points.

Approach	Grid Points	Time (s)	RMS (mm)
kd tree	-	2.62	0.013
Look-up table	$20 \times 20 \times 20$	0.004	0.046
Look-up table	$40 \times 40 \times 40$	0.004	0.029
Look-up table	$100 \times 100 \times 100$	0.004	0.021

the Kinect™, using the Bingham filter, to develop a point-cloud model of the environment. It is assumed that there is some overlap between the two point clouds. We demonstrate our approach on RGB-D images taken from the 'Freiburg1-Teddy' dataset of Sturm *et. al.* [205]. Fig. 6-11(a), (b) shows the snapshots of the images. Fig. 6-11(d) shows the final model of the room as generated by our approach. We use 20 simultaneous measurements and the same initial conditions as in the previous cases. Our approach takes ≈ 0.21s for estimating pose, which is twice as fast as ICP which takes ≈ 0.46s. In order to improve the speed we have implemented a C++ version of the Bingham filter, which takes only ≈ 2 ms [4]. The RMS error of our approach is 4.4cm, which is of the order of the accuracy of the sensor itself [204] and is better than the RMS error of ICP, 6cm.

Real-world Example– Stereo Point Cloud Registration and Tracking

In this section we look at an example of stereo image registration. The transformation between camera-frame and model-frame is estimated by registering the reconstructed point cloud from stereo images with the geometric model of the object.

Since the stereo camera images can consist of a number of objects, we first manually select the region containing the object of interest. We then refine the selection using a graph cut-based image segmentation. We use the variant of the BF that uses multiple simultaneous point measurements to estimate the pose. Since the pose estimation is fast, we repeatedly estimate the pose even after convergence. This allows us to track the object in the frame of the camera subject to slow motions. We use this

[4]Source code– https://github.com/biorobotics/bingham_registration/tree/ros-free

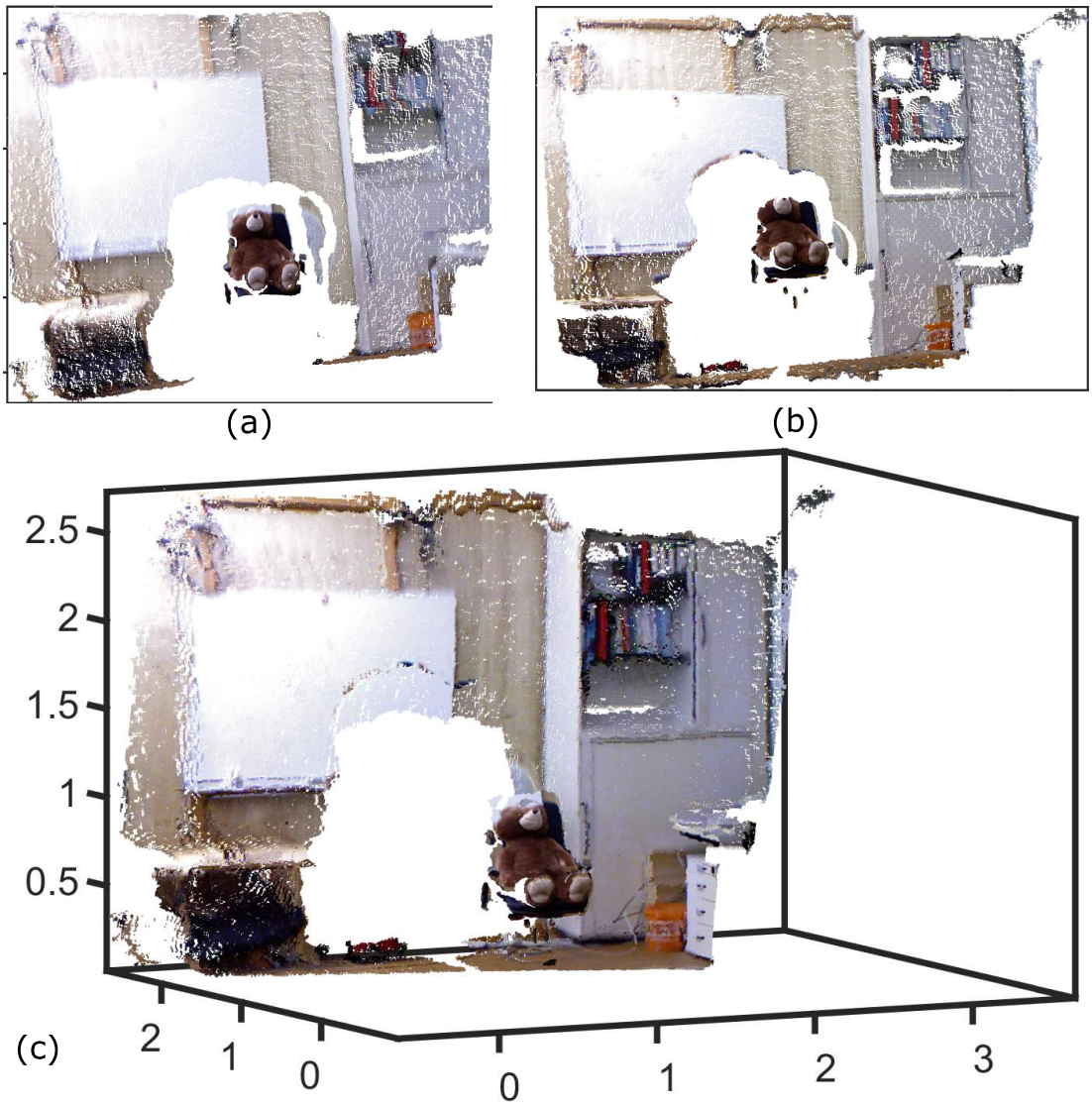


Figure 6-11: (a), (b), are two RGB-D images obtained from Kinect™, with some overlapping region.(c) The point cloud model estimated by aligning the point clouds in (a) and (b) using the Bingham filter. The BF takes 0.21s to estimate the pose with an RMS error of 4.4cm, as opposed to ICP, which takes 0.46s with an RMS error of 6cm.

approach to track 3 objects with varying level of geometric complexity. A Stanford bunny, a pelvis bone and a prostate. The bunny is geometrically most expressive and easy to track. The prostate is relatively symmetric and lacks interesting geometric features and is the most difficult of the three objects to track.

Fig. 6-12 shows the RMS error over a 60s time period of tracking, in which the bunny was moved by a user in arbitrary fashion. In the first 15 s, the user provides a

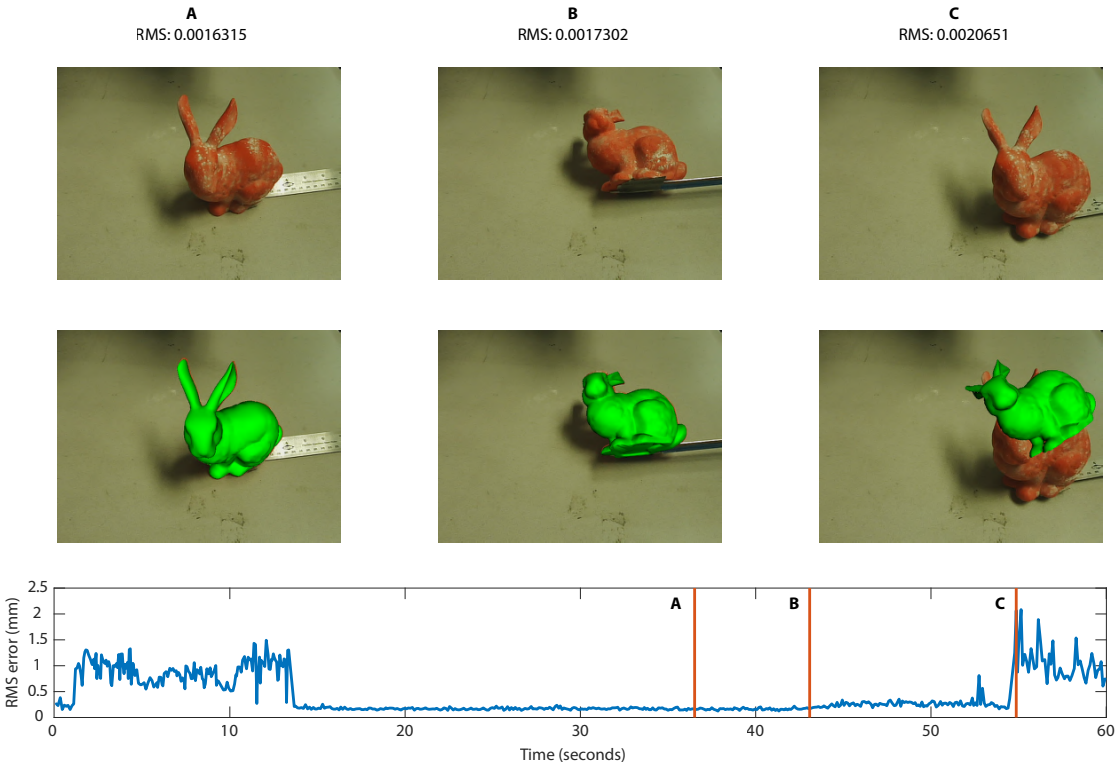


Figure 6-12: The top row shows the bunny at three time instances A, B and C as seen by the left stereo camera. The middle row shows the estimated pose of the bunny, in green, superimposed on the stereo image. The plot at the bottom shows the RMS error versus time, for pose estimation using Bingham filter, in an experiment that involved a user moving the bunny in the view of the camera.

region of interest in the stereo images and segments the object based on its hue and saturation. The BF then estimates the pose. Note that between 15-50 s, the bunny is being manually moved and our approach is able to robustly and accurately track the pose (For example instances A and B in Fig. 6-12). The RMS error is $< 2mm$, most of which can be attributed to the noise in the stereo point cloud. After about 55 s, the filter produces erroneous pose estimates (For example instance C in Fig. 6-12). This is because the bunny was moved very quickly and the incremental errors in the pose estimate resulted in a large difference from one frame to another and caused the point cloud segmentation to lose track of which points it was meant to be registering to.

Although the BF is able to handle moving objects, the image segmentation that we

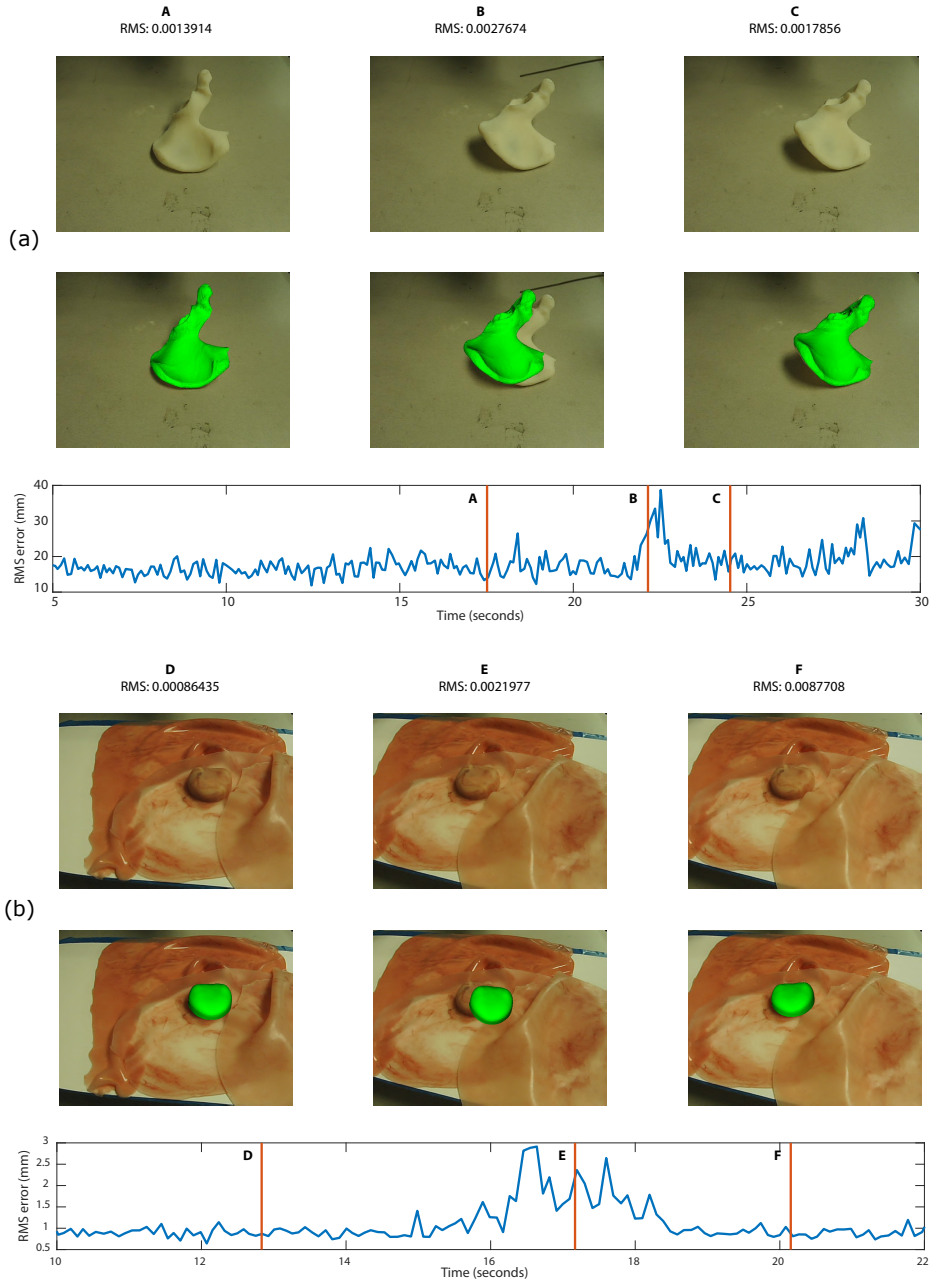


Figure 6-13: (a) Stereo image based tracking for a pelvis. Top row shows the left stereo camera image, middle row shows estimated pose superimposed and the bottom plot shows the RMS error vs time. At instance B, the pelvis was poked by the user which results in erroneous pose estimate. However, the pose estimate recovers to a low RMS error in a few seconds as shown in instance C. (b) Stereo image based tracking for a prostate. Top row shows the left stereo camera image, middle row shows estimated pose superimposed and the bottom plot shows the RMS error vs time. At instance E, the prostate is shifted by about 2 inches in less than a second, and the Bingham filter loses track of the object. However, the pose estimate recovers to a low RMS error in a few seconds as shown in instance F.

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

We repeat the experiments with 3D printed pelvis and silicone prostate (see Fig. 6-13). In the case of the pelvis, the RMS error in the first two seconds is around $1.5mm$. After this we apply a force on the pelvis in order to dislodge it from its location (see instance B in Fig. 6-13). Upon application of an external force on the object, the pose estimation becomes erroneous ($\approx 2.7mm$). However, the BF is able to quickly recover after the movement and within three seconds the RMS error has been reduced to $1.8mm$ (see instance C in Fig. 6-13). Similarly, in the case of the prostate, the RMS error is initially around $1mm$ (for example instance D in Fig. 6-13). We rotate and shift the prostate by about two inches in less than a second at around 17s (instance E in Fig. 6-13). The tracking RMS error increases to $2mm$, but it reduces to $0.8mm$ in about 3s.

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

future work.

6.4 Conclusion and Discussions

In this chapter, a Bingham distribution-based linear filter (BF) was developed for online pose estimation. Bingham distribution captures the bimodal nature of the distribution of unit quaternions as well as the unit norm constraint. By adapting the linear measurement model developed in Chapter 4, a linear Bingham filter has been developed that updates the pose based on a pair of position measurements. Further the filter is extended to process surface-normal, pose as well as multiple simultaneous measurements. We demonstrate the efficacy and the versatility of our approach on a number of application in simulation and real-world experiments. The applications include sequential point registration, hand-eye calibration, object tracking, registration of stereo-images and point-cloud stitching.

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

One drawback of our approach, as with most filtering based approaches, is that the estimate can be trapped in a local minima when the data association is unknown. Using a high initial uncertainty and more number of simultaneous measurements helps alleviate this problem to an extent. However, in some applications only pairs of measurements may be available per update, and the correspondences may be unknown (for example, probing-based registration). In such situations, better correspondences using a probabilistic metric as described by Billings *et. al.* [23], can improve the estimate. Another approach to resolve this issue is to use a global optimizer for filtering-based methods (see Chapter 7).

While the focus of this chapter was static pose estimation, we also demonstrated results for object tracking by running a series of static pose estimation. In the future, we plan to develop a process model to capture the dynamics of the moving object, and utilize an unscented Bingham filter [54] if this model is nonlinear. Another future direction involves using the estimate of the concentration matrix of the Bingham distribution to guide where to collect the next set of measurements that will improve the pose estimation.

6.5 Contribution

The contribution from this chapter is:

1. Development of a Bayes filter using a combination of Bingham and Gaussian distributions for fast, accurate and robust pose estimation. This approach can provide sequential pose estimates with serial or batch measurements.

2. Fast and accurate pose estimates using position, surface-normal and pose measurements.

3. A look-up table-based fast approach for dealing with unknown data association.

6.6 Published Work

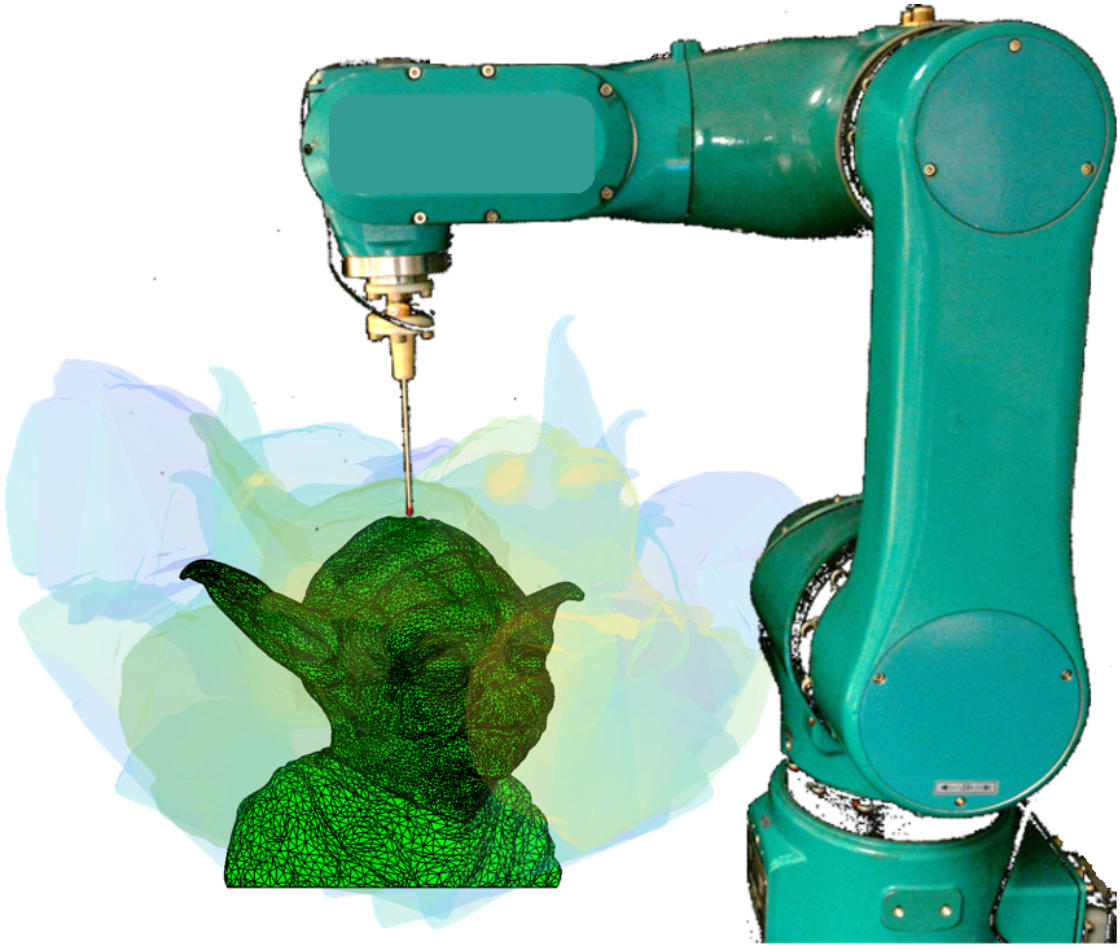
Material from this chapter has appeared in the following publication

1. R Arun Srivatsan, Mengyun Xu, Nicolas Zevallos and Howie Choset, "Bingham Distribution-Based Linear Filter for Online Pose Estimation, in the proceedings of Robotics: Science and Systems, Boston, USA, July 2017.

2. R Arun Srivatsan, Mengyun Xu, Nicolas Zevallos and Howie Choset, "Probabilistic Pose Estimation Using a Bingham Distribution- Based Linear Filter, International Journal of Robotics Research (IJRR), 2018.

Part III

Dealing with local minima in pose estimation



Chapter 7

Multiple Start Branch and Prune Filter

Chapter 4 and Chapter 6 discussed probabilistic approaches for pose estimation when the data association between the measurements were known. When data association is unknown, then results form those chapters do not guarantee global convergence, but only a local convergence. In this chapter we present a filtering approach for pose estimation when the data association in unknown. The approach we present is generic enough to be applied to other nonconvex optimization problems which are analytical and yet each function evaluation is expensive and have a relatively low dimensional (< 20) parameter space.

In various engineering applications such as automatic control systems, signal processing, mechanical systems design, image registration, etc., we encounter problems that require optimization of some objective function. While many efficient algorithms have been developed for convex optimization, dealing with nonconvex optimization remains an open question [206]. In this work, we introduce a new method for nonconvex optimization, called *multiple start branch and prune filtering algorithm* (MSBP). Compared to popular methods, branch and bound [207], simulated annealing [208], genetic algorithms [209], etc., MSBP only has a few parameters to tune and can provide fast online estimates of the optimal solutions.

We believe that Kalman filter-based methods for nonconvex optimization [80]

suffer less from issues surrounding computational efficiency and parameter tuning. Multi-hypothesis filtering initially proposed by Reid *et. al.* [84] and the heuristic Kalman algorithm (HKA) developed by Toscano *et. al.* [210, 80] are two popular choices for filtering based methods for nonconvex optimization. Both these methods, as well as MSBP, fall under the category of population based stochastic optimization techniques. MSBP was developed for nonconvex optimization problems where the objective function is available in an analytical form and yet is expensive to evaluate (for example the case of point registration).

Unlike the HKA which starts with one initial state estimate, MSBP starts with multiple initial states. These are further branched, updated and then pruned to explore the search space efficiently while avoiding premature convergence to a local minimum. A major advantage of MSBP over the methods mentioned above is the high success rate of estimating all the minima in problems with multiple local/global minima. The MSBP requires tuning of only three intuitive parameters, which makes it easy for a non-expert to use the method.

In this work we evaluate and compare the efficiency of MSBP to other methods on the Griewank function, which is a standard test for nonconvex optimization methods. We also test MSBP on point set registration. This application is specifically chosen to test our algorithm because of its analytical and yet expensive function evaluation which offers practical challenges to most of the existing algorithms for nonconvex optimization. MSBP is tested in the presence of high initial error, multiple global minima, noisy data and incomplete data. In all these cases, MSBP accurately estimates the global minima with a high success rate over multiple runs of the algorithm.

7.1 Related Work

In a general setting, an optimization problem consists of finding input variables within a valid domain that minimize a function of those variables. An optimization problem

can be represented as

$$\begin{aligned} \text{minimize} \quad & h(\boldsymbol{x}), && \boldsymbol{x} \in \boldsymbol{R}^{n_{\boldsymbol{x}}} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, && i = 1, \ldots, n_c \\ & e_j(\boldsymbol{x}) = 0, && j = 1, \ldots, n_e. \end{aligned} \tag{7.1}$$

In Eq. 7.1, $\boldsymbol{x}$ is the $n_{\boldsymbol{x}}$ dimensional input variable, also known as the optimization variable, h is the objective function to be minimized, $g_i(\boldsymbol{x})$ and $e_j(\boldsymbol{x})$ are the inequality and equality constraints respectively and n_c and n_e are the number of inequality and equality constraints respectively.

Nonconvex optimization problems

We often encounter optimization problems that have a number of locally optimal solutions which are optimal only within a small neighborhood but do not correspond to the globally optimal solution that minimizes the function in the function domain. Such problems are termed "nonconvex" optimization problems, in contrast to "convex" optimization problems where any local minimum is also a global minimum. Nonconvex optimization problems are in general non-trivial to solve because it is difficult to guarantee that the solution returned by the optimizer is global rather than local.

For these problems, a standard approach is to use convex optimizers that employ different randomization techniques to choose multiple initial starts, for example the approach of Schoen *et. al.* [211]. The drawback of this approach is that for problems with a large number of local minima solutions, a lot of computational effort may be needed to find the global optimum (see [206] for a detailed discussion on this). Branch and bound methods are also commonly used, but the curse of dimensionality leaves them ineffective in cases with many optimization variables as pointed out by Lawler *et. al.* [207].

Heuristic methods for nonconvex optimization problems

Several heuristic methods have been developed to estimate global minima in nonconvex optimization problems such as simulated annealing (SA) [208], particle swarm optimization (PSO) [212], genetic algorithms (GA) [209] and more recently recursive decomposition (RD) developed by Friesen and Domingos [213]. SA is widely considered as versatile and easy to implement, but there are two major drawbacks: 1) there are multiple unintuitive parameters that require tuning, and the results are known to be sensitive to the choice of these parameters (The sensitivity to parameters and difference between practice and theory of SA has been well studied by Ingber [214]); 2) the computation time is generally high for most practical applications and often parallel is required, such as the approach of Ram *et. al.* [215]. PSO and GA are both categorized as population-based random-search methods. PSO is more sensitive than GA to the choice of parameters, and is known to prematurely converge unless the parameters are tuned well. Also, GA is known to be computationally intractable for many high dimensional problems, as observed by Seixas *et. al.* [83]. In contrast, RD decomposes the objective function into approximately independent sub-functions, and then optimizes the simpler sub-functions using gradient based techniques. The drawback of such a method is that not all functions can be decomposed into sub-functions, in which case RD would perform similarly to a gradient descent with multiple starts.

Filtering-based methods for nonconvex optimization problems

Due to their ease of use and small number of tuning parameters, Kalman filter-based methods have also been used in optimization [216, 217, 210]. Typically such methods adapt a Kalman filter to have a static process model with the state vector comprised of the optimization variables $\boldsymbol{x}$ and an initial state uncertainty $\boldsymbol{\Sigma}$ spanning the domain of the search space. The measurement model is taken to be an evaluation of the objective function. The measurement is chosen to be the value of the minimum that we want the objective function to attain. By definition, with each iteration of the Kalman filter, the state vector is updated such that the difference between the true

measurement and the expected measurement is decreased [51], thus ensuring that the objective function is minimized. The corresponding covariance also decreases as the number of iterations increases. When the mean of the state stops changing over iterations, or when the uncertainty decreases below a set threshold, we consider the state to be the optimal estimate.

As shown in Sec. 2.6, a Kalman filter can provide the optimal estimate of $\boldsymbol{x}_{k|k}$ [1] such that

$$\boldsymbol{x}_{k|k} = \underset{\boldsymbol{x}}{\operatorname{argmax}}\ p(\boldsymbol{x}|\boldsymbol{z}_k, \boldsymbol{x}_{k|k-1}), \tag{7.2}$$

provided $\boldsymbol{x}$ and $\boldsymbol{z}$ are both Gaussian distributed and the likelihood $p(\boldsymbol{z}_k|\boldsymbol{x}_{k|k-1})$ is linear in $\boldsymbol{x}_{k|k-1}$. The solution to Eq. 7.2 is derived in Sec. 2.6,

$$\begin{aligned}\boldsymbol{x}_{k|k} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\ &(\boldsymbol{x} - \boldsymbol{x}_{k|k-1})^T \boldsymbol{\Sigma}_{k|k-1}^{-1} (\boldsymbol{x} - \boldsymbol{x}_{k|k-1}) + \\ &(\boldsymbol{z}_k - h(\boldsymbol{x}))^T \boldsymbol{R}_k^{-1} (\boldsymbol{z}_k - h(\boldsymbol{x})),\end{aligned} \tag{7.3}$$

where $\boldsymbol{\Sigma}_{k|k-1}$ and $\boldsymbol{R}_k$ are the uncertainties associated with $\boldsymbol{x}_{k|k-1}$ and $\boldsymbol{z}_k$ respectively. Note that Eq. 7.3 contains a nonlinear function $h(\boldsymbol{x})$ instead of the linear model $\boldsymbol{H}\boldsymbol{x}$ as in Eq. 2.31. h is the unconstrained objective function as defined in Eq. 7.1.

Let h_{min} be the smallest value that h can attain, which is attained at $\boldsymbol{x} = \boldsymbol{x}_{min}$. Since there is uncertainty associated with $\boldsymbol{x}$, we have $h_{min} = h(\boldsymbol{x}_{min}) = h(\boldsymbol{x}) + \boldsymbol{v}$, where $\boldsymbol{v} \sim \mathcal{N}(0, \boldsymbol{R}(\boldsymbol{x}))$ is state dependent measurement noise drawn from a zero mean distribution with covariance $\boldsymbol{R}$. For an optimization problem as shown in Eq. 7.1, we choose the following measurement model

$$\boldsymbol{z}_k = h(\boldsymbol{x}) + \boldsymbol{v}_k(\boldsymbol{x}).$$

We set the measurement $\boldsymbol{z}_k = h_{min}$, as the state $\boldsymbol{x}$ will then be updated such that the value of h is close to h_{min}. If h_{min} is not known *a priori*, $\boldsymbol{z}_{k+1}$ can be set to an arbitrarily small value. The uncertainty $\boldsymbol{R}(\boldsymbol{x})$ can be computed analytically as

[1] $\boldsymbol{v}_{a|b}$ is the estimate of $\boldsymbol{v}$ at the a^{th} iteration given measurements upto b iterations.

shown in [146, pp. 90–91]. The resulting filter would provide an optimal estimate of $\boldsymbol{x}$ as long as h is linear [51], as shown in Sec. 2.6. The following are the Kalman filter equations modified for an optimization problem:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{K}_{k+1}(h_{min} - h(\boldsymbol{x}_k)), \tag{7.4}$$

$$\boldsymbol{\Sigma}_{k+1} = \boldsymbol{\Sigma}_k - \boldsymbol{K}_{k+1}\boldsymbol{H}_{k+1}\boldsymbol{\Sigma}_k, \tag{7.5}$$

where the Kalman gain $\boldsymbol{K}_{k+1} = \boldsymbol{\Sigma}_k\boldsymbol{H}_{k+1}^T(\boldsymbol{H}_{k+1}\boldsymbol{\Sigma}_k\boldsymbol{H}_{k+1}^T + \boldsymbol{R})^{-1}$ and $h(\boldsymbol{x}_k) = \boldsymbol{H}_k\boldsymbol{x}_k$. $\boldsymbol{H}$ is the Jacobian of the objective function $h(\boldsymbol{x})$.

If h is nonlinear, variants such as the EKF, UKF, etc. can be used. In the presence of constraint functions that must be satisfied as shown in Eq. 7.1, equality or inequality constrained Kalman filtering techniques can be applied. The works of Simon and Chia [218] deal with equality constrained filtering, while Gupta and Hauser [217] deal with both equality and inequality constrained filtering and Tully *et. al.* [47] provide an improved approach to deal with inequality constrained filtering.

In general, the Kalman filter only estimates the local minimum. Popular approaches for nonconvex optimization problems include multi-start and multi-hypothesis filtering as shown in Fig. 7-1(a) (originally introduced by Reid [84]). Multiple filters each having a different randomly chosen initial start, are run in parallel, and after each iteration the estimate with the maximum likelihood is chosen as the current best estimate. Such an implementation has a good chance of finding global minima but at the expense of increased computation time.

Particle filters have also been adapted as a smart alternative to multi-hypothesis filtering, such as the work of Ma *et. al.* [16]. The resampling step in a particle filter ensures that states with low weights are pruned while the others are retained (see Fig. 7-1(b)). Particle filters and multi-hypothesis filters both suffer from the curse of dimensionality. When estimating four or higher dimensional parameters, a large number of particles are needed to span the search space to find the global optimum, which can be computationally expensive especially if the function evaluation is not cheap.

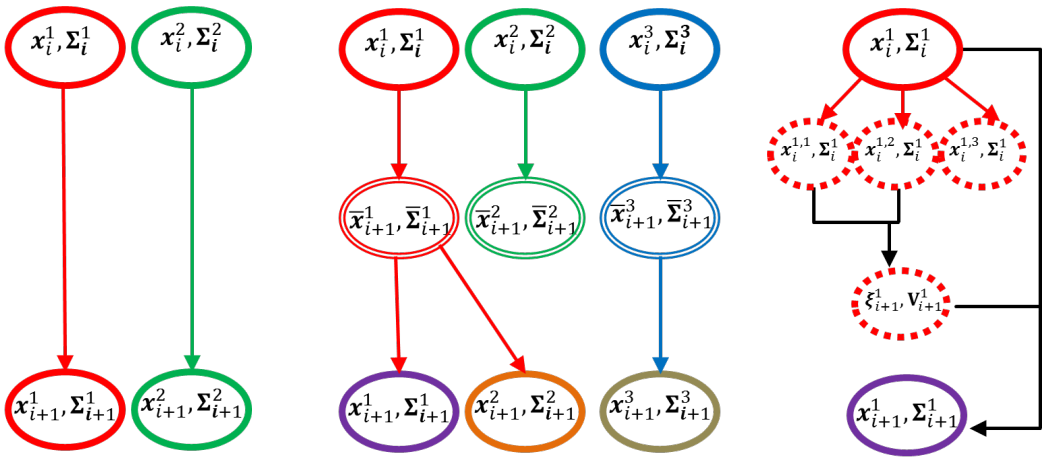


Figure 7-1: (a) Steps involved in one iteration of a multi-hypothesis filter with 2 initial start states. After each iteration the state with maximum likelihood estimate is chosen as the best current estimate. (b) Steps involved in a particle filter with 3 particles. After updating the particles based on the measurement, resampling is performed to remove particles with low weights. (c) Steps involved in one iteration of the heuristic Kalman algorithm. In this example, the parent's state is divided into 3 child states. The weighted sum of 2 child states with the lowest objective value is used to obtain the pseudo measurement $\boldsymbol{\xi}_{i+1}$.

The HKA, introduced by Toscana *et. al.* [80], is a combination of Kalman filtering and population-based random-search methods (see Fig. 7-1(c)). Starting with a parent state, HKA spans child states and evaluates the function at the child states. A pseudo measurement and its uncertainty $(\boldsymbol{\xi}_{i+1}, \boldsymbol{V}_{i+1})$ are then obtained from the n best states with the smallest function value, and the state $(\boldsymbol{x}_i, \boldsymbol{\Sigma}_i)$ is updated using the pseudo measurement. Even though the parent state is divided into a number of child states, in each iteration of the algorithm only a single state, is updated. Such an approach has been shown to be suitable in situations where the function can only be evaluated using experimental simulations and not analytically. For such problems, HKA is a good optimization tool with very few parameters to tune and a good success rate of finding global minimum [210]. However, when an analytical form of the objective function is available, other methods perform much better than HKA.

7.2 Problem Formulation

In this section we show the formulation of our approach, multiple start branch and prune filter (MSBP). The basic framework of the MSBP is shown in Fig. 7-2. The various steps involved in the MSBP implementation are as follows:

1. The algorithm is initialized with n initial parent states $(\boldsymbol{x}_k^i, \boldsymbol{\Sigma}_k^i)$, $i = 1, 2, \ldots, n$, where k denotes the iteration index (see Section 7.2.1 for information on how to choose the initial states).

2. Each parent state is divided into m child states $(\boldsymbol{x}_k^{i,j}, \boldsymbol{\Sigma}_k^i)$, $(j = 1, 2, \ldots, m)$, by sampling from the distribution $(\boldsymbol{x}_k^i, \boldsymbol{\Sigma}_k^i)$. The parent state is always retained as one of the m child states. The child states that are generated from the parent state can be viewed as perturbations being added to the states to overcome local minima and to encourage exploration.

3. The child states are then updated using Eq. 7.4 to obtain $(\boldsymbol{x}_{k+1}^{i,j}, \boldsymbol{\Sigma}_{k+1}^{i,j})$.

4. From the $n \times m$ child states, the n states with the lowest innovation, i.e., $\boldsymbol{z}_{k+1} - h(\boldsymbol{x}_{k+1|k})$ from Eq. 7.3 are chosen as parents for the next iteration.

5. Among the n parent states chosen, if the means of any states come within an ϵ threshold of each other, the state with the lower innovation is retained and the others are pruned (n decreases every time pruning happens).

6. Steps 2-5 are repeated until convergence or up to a fixed number of iterations.

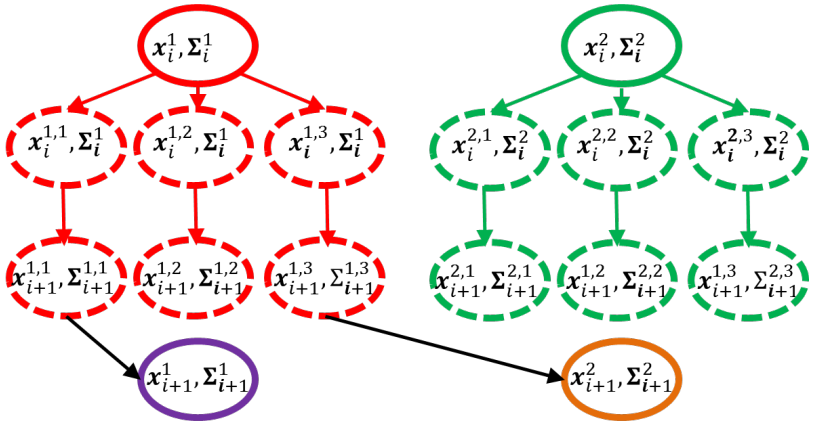


Figure 7-2: Steps involved in one iteration of the MSBP. Parent states are shown in bold ellipses and child states are shown in dashed ellipses. In this example, $n = 2$, $m = 3$.

From Fig. 7-1(a) it can be noted that multi-hypothesis filtering is a special case of MSBP with $m = 1$ and $\epsilon = 0$. The multi-hypothesis filter requires a large number of initial states to converge onto a global minimum, as a lack of perturbation can result in premature convergence to a local minimum. Also lack of a pruning step in the multi-hypothesis filter often results in duplication of estimates by multiple filters. Particle filters prune states with lower probability during the resampling step and offer an advantage over multi-hypothesis filter. However, particle filters lack the perturbation step and the state update step present in MSBP, which helps in over coming local minima and quick convergence to the optimal solutions (Though we do not have a proof for this, we empirically observe this in our experiments). In comparison to other methods, such as SA, GA, PSO, etc., at each iteration in addition to evaluation of the objective function at multiple states, the states themselves are updated by the update model of the MSBP. While this could be viewed as additional computation, the update step allows us to minimize the function faster and quickly

identify the minima compared to the other methods. MSBP provides a maximum of n estimates after each iteration as opposed to a single estimate provided by HKA (see Fig. 7-1(c)). This is a drawback for HKA in problems that have multiple global minima, as HKA would tend to return an estimate that is at a location intermediate to both the minima. Running the HKA multiple times with different initial states can improve the success rate of finding the global minimum, but at the cost of increased computation time.

Thus, the MSBP offers the advantage of reduced computational load and memory storage in addition to a higher success rate of estimating the global minima, for problems with analytical objective functions. The only shortcoming is that when dealing with very high dimensional systems (typically > 20), the update step of the Kalman filter can become expensive as it would involve inverting a high-dimensional matrix.

7.2.1 Choice of Initial State and Parameters

In addition to the choice of initial states, there are three parameters that require tuning in the MSBP: n, m and ϵ. This section describes the intuition behind selecting these parameters and the initial states.

- Initial state: In most practical problems the domain of the search space for optimization is known. Without loss of generality, the uncertainty of all the initial states is chosen to be equal to each other. The uncertainty is chosen to be a diagonal matrix with each diagonal element set to be equal to σ^2, such that 6σ equals the span of the domain in that dimension. Such a choice for $\mathbf{\Sigma}_0^i$ is generally conservative, and restricts the uncertainty in each of the parent state to the search domain. The mean of the states $\boldsymbol{x}_0^i$ are randomly chosen from the valid search domain.

- Number of parent states n: n can be chosen based on prior knowledge of the number of global minima present in the problem. If that number is not known *a priori*, then a conservative estimate can be made. In practice we observe that

choosing a value of n greater than the number of global and local minima present in the search domain improves the success rate of the algorithm. However, increasing the value of n also increases the computation time.

- Number of child states m: m is the number of child states per parent state. If the estimator is stuck at a local minima, the perturbations help get it out of the local minima. Hence, the greater the value of m, the greater the chances of MSBP capturing the global minima. However a higher m would also mean increased computation time. As result m should be chosen depending on the allowable computation time for the application.

- Choice of ϵ: ϵ is the parameter that decides the threshold between the parent states. ϵ helps prevent unnecessary computation and encourages exploration. A large value of ϵ can prune several parent states at once and can result in missing some solutions. $\epsilon = 0$ would not prune any parent state, resulting in unwanted computation in cases where multiple parent states are identical. Depending on the application, ϵ can be chosen to be a fixed number for all iterations or its value can be varied over the iterations.

Section 7.3.2 describes in more detail how these parameters are tuned for a case study on point set registration.

7.3 Results

In this section, we first demonstrate the performance of MSBP by testing it on the Griewank function. Following that, we do a case study of point set registration problem.

7.3.1 Numerical experiment with Griewank function

A number of standard functions are used to test the performance of nonconvex optimization methods [219]. In this work, we choose to test the MSBP on the Griewank

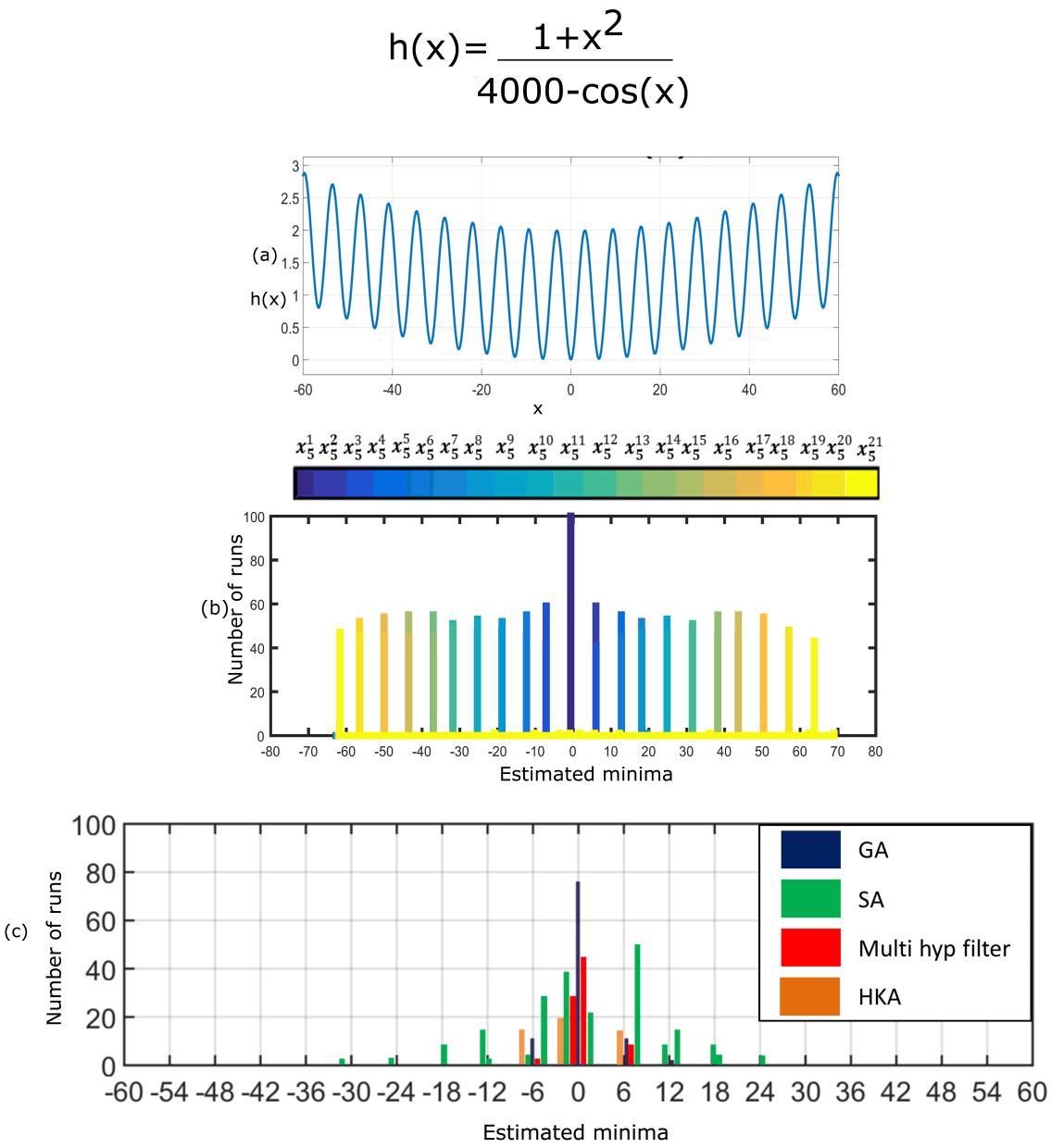


Figure 7-3: (a) A plot of the Griewank function. (b) A histogram showing the values estimated by 21 parent states of MSBP over 100 runs. The Y axis of the plot shows the number of runs that estimate a particular state and the X axis shows the estimated value. A histogram of the estimated value over 100 runs is shown for the following algorithms (c) Histogram showing values estimated by Genetic algorithm, Simulated annealing, Multi-hypothesis filter, and HKA.

function. Fig. 7-3(a) shows the plot of Griewank function for $x \in [-60,\ 60]$. In the chosen domain, the function is known to have a global minima at $x = 0$ and twenty local minima at ± 6.28, ± 12.56, ± 18.84, ± 25.12, ± 31.45, ± 37.55, ± 43.93, ± 50.3, ± 56.67.

As mentioned in Section 7.2.1 in order to ensure that 99% of samples fall within the search domain we choose the uncertainty of the initial states, $\Sigma_0 = \sigma_0^2$, such that $6\sigma_0 = 120$. The mean of the initial parent states are sampled from the normal distribution $\mathcal{N}(\mu, \Sigma_0)$, where $\mu = 5$. We choose $\mu = 5$, as it is closer to the local minima at $x = 6.28$ than the global minima at $x = 0$, and would be a more challenging test for the optimization algorithm. For our implementation of MSBP, we use an EKF since the function is non-linear. In addition we choose $n = 21, m = 10, \epsilon = 2$. We run all the algorithms until convergence. The algorithm is set to have converged when the change in the estimate of the minima is less than 10^{-6}. We observe that the maximum number of iterations required by any algorithm is generally under 20. For the sake of a fair comparison, the values of the parameters for all the algorithms were tuned as per the recommendation in [80] and the best results have been reported.

We repeat the experiment 100 times to observe the performance of the method over multiple runs. Fig. 7-3(b) shows the histogram of the values estimated by MSBP over 100 runs, all of which converged within five iterations. The global minimum is estimated correctly at $x = 0$ each time, while the local minima solutions are accurately predicted by the remaining twenty parent states. The order in which the other parent states estimate the local minima varies in each run of the algorithm, but they are tracked in all of them.

In comparison, HKA was implemented with initial state $(x_0, \Sigma_0) = (5, 400)$, 20 divisions, 2 best candidates, and a slow down coefficient of 0.7 . Fig. 7-3(c) shows the histogram of values estimated by HKA over 100 runs. We observe that the algorithm correctly estimates the local minima only 10% of the time. In 8% of the runs, HKA estimates the local minima at $x = 6.28$ and $x = -6.28$, which are closest to the mean of the initial state, $x_0 = 5$. SA also estimates the global minimum only 15% of times and rest of the times it gets stuck at nearby local minima (see Fig. 7-3(c)).

A multi-hypothesis filter was also implemented, where we choose the same initial states as those for MSBP with $n = 21, m = 1, \epsilon = 0$. Fig. 7-3(c) shows the histogram of estimated values over 100 repeated runs. More than 50% of the time, the algorithm estimates the global minimum correctly. The rest of the times it estimates values close to the global minimum or one of the two local minima closest to the global minimum similar to the GA (Fig. 7-3(c)).

7.3.2 Rigid Registration

Rigid registration is the process of finding a spatial transformation that aligns the elements of two point sets. When the correspondence between the points in the two point sets is known, rigid registration can be solved analytically as shown by Horn [4], Arun *et. al.* [17], Faugeras [60] and Walker *et. al.* [61]. However, when point correspondences are unknown, finding the optimal transformation becomes challenging due to the presence of several local minima solutions. Besl and McKay [11] came up with the popular ICP method that recursively finds correspondences and minimizes the alignment difference between point sets. But this is only a locally optimal approach, and hence it is critical to start with a good initial condition. A branch and bound based technique, called Go-ICP, has been developed to avoid this problem by Yang *et. al.* [18]. However, this method has high computation time and is not suitable in real time applications. In this work, we use the MSBP for registration of point sets and demonstrate that it is able to find accurate estimates with low computation time. We perform multiple case studies with different conditions using different standard 3D shape datasets to show the versatility of our algorithm.

We use a dual quaternion-based filter (DQF) for estimating the registration parameters (see Chapter 4). The MSBP can also be used with other filtering implementation for registration estimation such as the UKF of Moghari *et. al.* [5] and EKF of Kang *et. al.* [220]. The DQF uses dual-quaternion representation for pose, and then reposes the originally nonlinear estimation problem as a linear estimation problem. This allows us to use Eq. 7.4 for estimating the optimal registration parameters (see Chapter 4 for more information on the expression for objective function used). In

each iteration of the MSBP, closest point correspondence is found between a pair of points (as opposed to finding correspondence for all the points in the case of methods such as ICP). The correspondence found using the parent states is retained for the child states as well. Since the number of different correspondences that can be formed between the two point sets is combinatorial, we expect many local minima solutions. Hence, we choose a large value for n in all the applications below. When the state uncertainty reduces below a desired threshold, we end the estimation process and stop collecting measurements. Thus, the DQF has a built-in stop condition that allows you to process a smaller number of measurements compared to batch processing methods such as the ICP, in which we wait for all the measurements to be collected before estimating the optimal registration.

Large initial transformation error

Fig. 7-4(a) shows the CAD model of a Stanford bunny [7]. The CAD model is geometrically discretized using a triangular mesh with 43318 triangle vertices. We collect 1000 random samples of points from the CAD model and apply a known transformation to those points (see Table 7.1 for the applied transformation). We then estimate the applied transformation with the MSBP. The values of various parameters used are $n = 40, m = 10, \epsilon = 1$. The experiment is repeated 100 times to note the statistical performance of our method. Fig. 7-4(a) shows the MSBP estimated points lie on top of the CAD model indicating accurate registration. On an average our algorithm converges after using 120 measurements.

Table 7.1 shows the actual registration parameters and the estimated values. The algorithm is compared with HKA, multi-hypothesis filtering, ICP, SA and GA. The SA and GA implementation we use for the sake of comparison are as described by Luck *et. al.* [82] and Seixas *et. al.* [83], which are a modified form of the original implementations of SA and GA with ICP running internally in each iteration. Luck *et. al.* [82] and Seixas *et. al.* [83] show that even though their approaches are expensive per iteration, they result is requirement of fewer iterations over all for convergence, and hence are faster and more accurate at estimating the registration

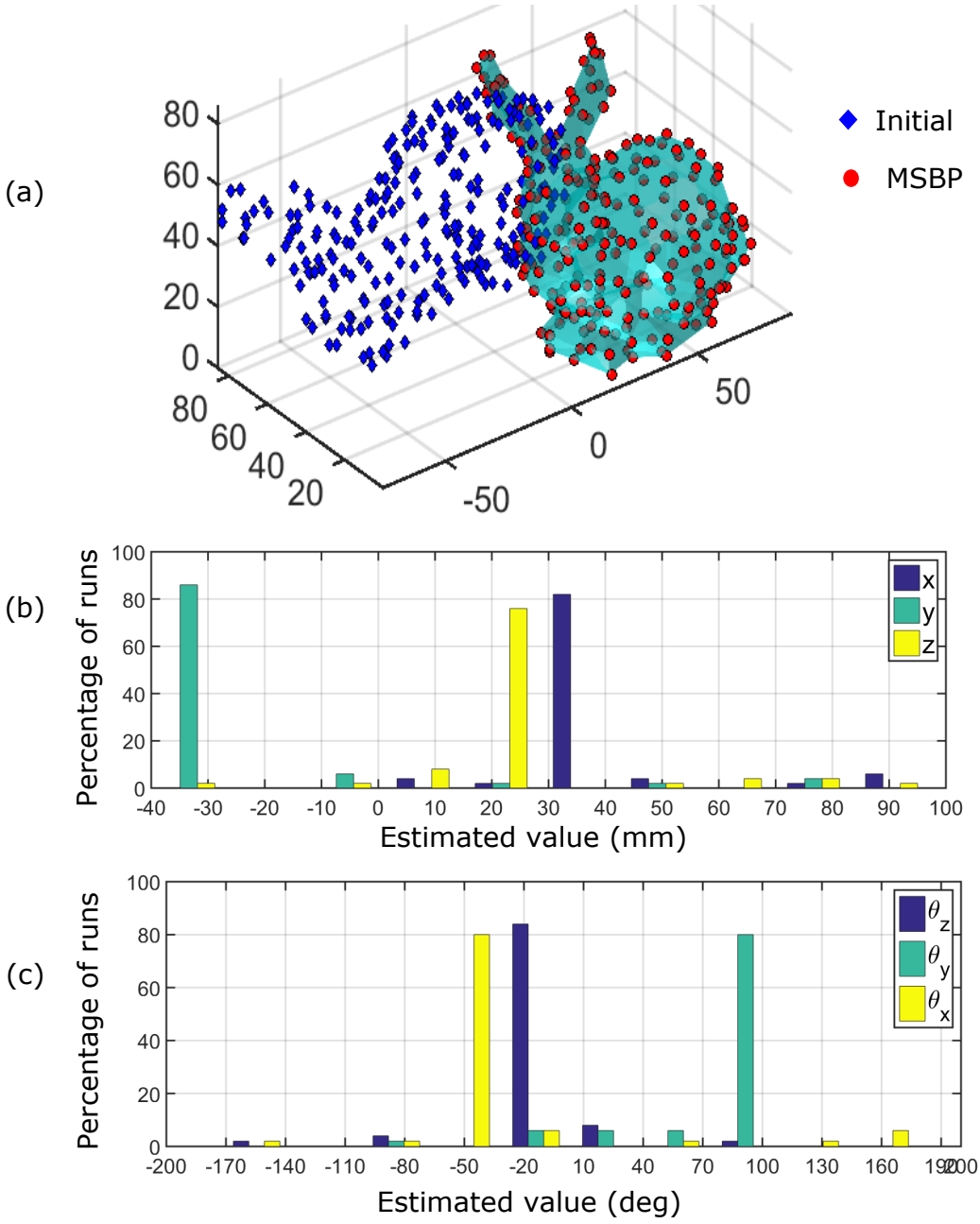


Figure 7-4: (a) CAD model of a Stanford bunny. The initial position of 1000 points in shown in blue-diamond markers, the position estimated by MSBP is shown in red-circular markers. (b) Histogram of the estimated translation parameters, (c) histogram of the estimated rotation parameters over 100 runs of the algorithm. In (b) and (c), the Y axis shows the percentage of runs that return a particular value and the X axis shows the estimated value returned by the parent state with the smallest innovation. MSBP has a high success rate of estimating the optimal parameters.

Table 7.1: Comparison of pose parameters as estimated by different registration methods for a case with large initial transformation error

Case1	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)	Time (sec)
Actual	30	-40	15	-55	80	-20	–	–
MSBP	29.89	-39.84	14.67	-58.57	80.59	-23.31	0.48	28
ICP	42.04	-35.22	8.52	17.83	19.21	33.26	35.06	5.82
Multi-hyp	59.79	-20.66	15.26	53.08	-45.58	30.29	18.25	404.62
HKA	-3.97	-17.69	17.45	31.39	31.69	-22.05	53.44	201.97
SA	29.16	-38.34	13.97	-51.75	81.67	-14.49	2.36	353.67
GA	30.08	-39.93	15.05	-54.59	79.92	-19.51	0.08	1051.00

parameters (These observations have been independently verified by us and hence we do not report results for vanilla implementations of SA and GA in this work).

While MSBP and multi-hypothesis filter estimate the registration parameters in a dual-quaternion space, we convert the estimated values into Cartesian coordinates and Euler angles for easy comparison with other methods. The penultimate column and the last column of Table 7.1 show the RMS error and time taken for various algorithms [2].

For multi-hypothesis filtering, we use the same set of initial states as MSBP. For HKA, ICP and SA we use a 4×4 identity matrix as the initial transformation. The bounds on the search space are $[-100,\ 100]$ for translation and $[-\pi,\ \pi]$ for rotation around each axis. For HKA we use 40 divisions, 4 best candidates and a slow down coefficient of 0.4. For GA we use an initial population of 100, cross-over probability of 0.7 and mutation probability of 0.2. These values for the parameters are tuned as per [210] and the best results are reported.

Fig. 7-4(a) shows that the displacement between the initial position of the points and their true position on the CAD model, is quite high and ICP does not perform well for such high initial errors. We notice that HKA also does not estimate the transformation accurately, presumably because it gets stuck at a local minimum. MSBP, SA and GA accurately estimate the transformation, however, SA and GA

[2]The computational time taken is calculated for script written in MATLAB R2015a software from MathWorks, running on a ThinkPad T450s (20BX0011GE) laptop from Lenovo with 8 GB RAM and intel i7 processor.

take much more time than MSBP to estimate. Since each function evaluation consists of an iteration of ICP internally, SA and GA both have higher estimation time than MSBP.

The MSBP algorithm is run 100 times and a histogram of the estimated translation and rotation parameters are shown in Fig. 7-4(b) and Fig. 7-4(c), respectively, which show that there is a $> 85\%$ chance of MSBP converging to the correct value. In comparison with MSBP, GA has a success rate of 10% and SA has a success rate of 20%. Thus, the MSBP produces accurate and repeatable results with high success rate, despite large errors in initial registration.

Multiple global minima

In this example, we consider a snowflake as shown in Fig. 7-5(a), which has rotational symmetry about an axis passing through its center and perpendicular to its plane. The object is symmetric to its original shape when rotated about this axis by $\pm 60°, \pm 120°$ and $180°$. We sample 100 points from the CAD model of the snowflake and transform those points by a known transformation: $(x, y, \theta_z) = (15\text{mm}, 30\text{mm}, 45°)$. We then use MSBP to estimate the applied transformation. Since the snowflake is 2-dimensional, we restrict ourselves to in-plane registration. We use the following parameter values for MSBP: $n = 100, m = 10, \epsilon = 5$. After 100 iterations, the number of surviving parent states is 16. Fig. 7-5(b)-(g) show the position of the points after applying a transformation given by the first six parent states as estimated by the MSBP. The first six parent states of the MSBP accurately capture the six global minima (Note that we limit our search domain to $[-180°,\ 180°]$ and hence there are 6 global minima in the search domain upto the rotational periodicity).

Noise in the input data

In order to test the robustness of the registration using MSBP in the presence of noise in one of the point sets, we consider the example of Fertility as shown in Fig. 7-6. 200 Points are sampled from the CAD model and a Gaussian noise $\mathcal{N}(0, \sigma_n^2)$ is

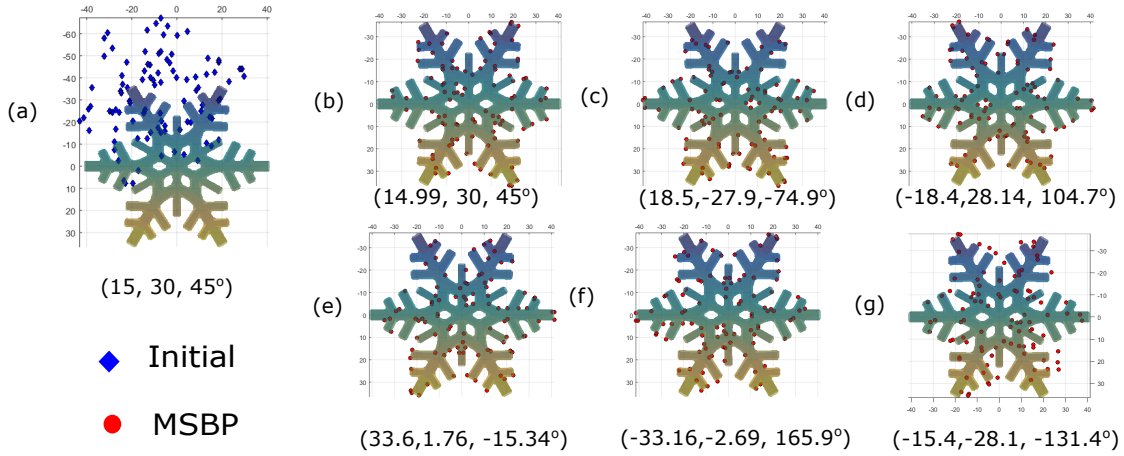


Figure 7-5: (a) CAD model of a snowflake.The initial position of 100 points and the position estimated by MSBP are shown in blue-diamond and red-circular markers respectively. The actual transformation between the points and the CAD model is $(15, 30, 45°)$. (b)-(g) The first six parent states of MSBP. The estimated registration parameters are given below the figure. Note how the rotation angles are $45° \pm n \times 60°$, $(n = 0, 1, 2)$ due to the 6 way symmetry in the shape of the snowflake. Snowflake CAD model courtesy of Thingiverse CAD model repository

applied to each of the points. The standard deviation σ_n is kept constant for all the points, but is gradually increased from 1 to 20 in increments of 1 over several runs (For reference, the CAD model can be fit in a box of size $300 \times 200 \times 100$ units). Left hand side of Fig. 7-6(a)-(c) shows that CAD model and the initial position of the points in blue-diamond markers for 3 different values of σ_n. The right hand side of Fig. 7-6(a)-(c) shows the CAD model and the location of the points after applying the transformation as estimated by MSBP.

Note how the MSBP is able to successfully register the points for all the three cases shown in the figure. Also note how after registration, the points appear to be lying on the CAD model for lower σ_n and appear to be spread out of the CAD model for the case with higher σ_n.

We do not present any systematic study to deal with outliers in this chapter. However, we empirically observe that our approach is able to perform reasonably well when there is about 15% outliers in the measurements. Table 7.2 shows the average RMS error over 100 experiments for the case of registering 500 randomly sampled points on a Stanford bunny (The model can fit in a box of size 85mm $\times$

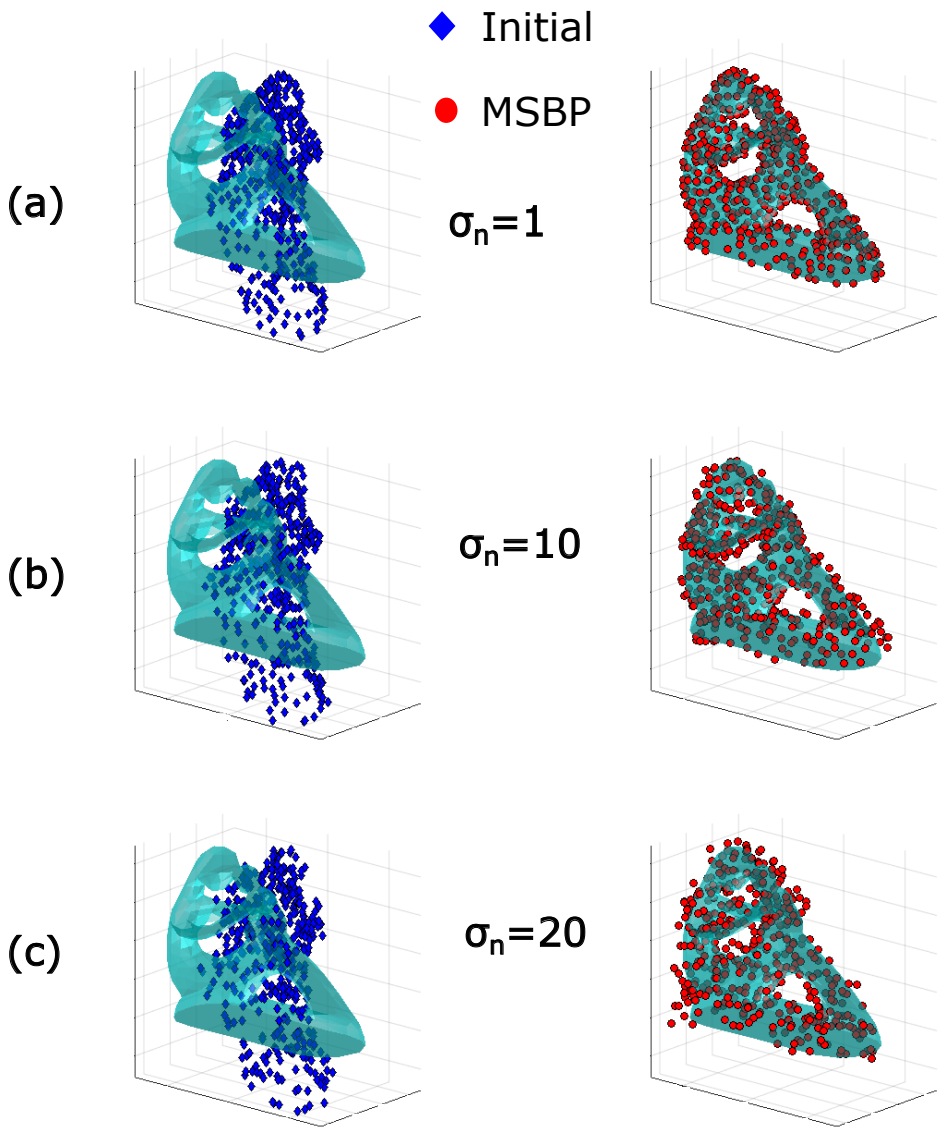


Figure 7-6: CAD model of Fertility and 100 points sampled from it and a noise $\mathcal{N}(0, \sigma_n^2)$ is added to the points. (a) the plot for $\sigma_n = 1$ (b) the plot for $\sigma_n = 10$ (c) the plot for $\sigma_n = 20$. CAD model of Fertility courtesy of AIM@SHAPE model repository

85mm $\times$ 85mm), which has various levels of measurement outliers. From the available measurements, we randomly pick a chosen percentage of measurements. We then add to the chosen measurements a noise which is randomly obtained from $[-100,\ 100]$mm along each axis, to simulate outliers. Note that if the RMS error is greater than 35% of the size of the object, then we consider that estimation result to be a failure. Table 7.2 also shows the success rate for each case. Note that failed estimates are

not included in the calculation of average RMS errors. As we increase the percentage of outliers, the performance of our approach gets worse. The success rate drops and the RMS error increases. To deal with outliers, one could follow the approach of Myronenko *et. al.* [116], Billings *et. al.* [221] or Izatt *et. al.* [20].

Table 7.2: Registration with MSBP in the presence of outliers

Outlier %	RMS (mm)	Success (%)
0	1.01	100
10	1.43	80
15	2.72	80
20	5.95	70

Robustness to incomplete data

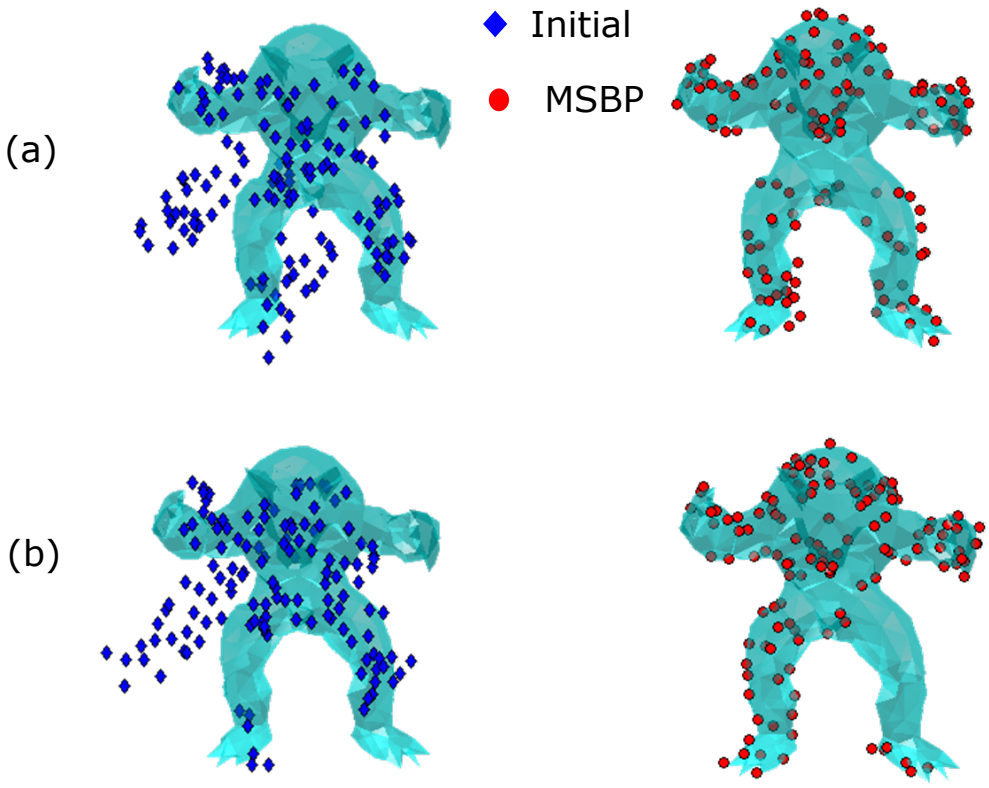


Figure 7-7: CAD model of a Stanford Armadillo man [7] and set of initial points sampled from parts of the model. The points are not sampled uniformly from all over the CAD, but have regions of missing information. (a) and (b) show two instances of incomplete data registered accurately to the CAD model using MSBP.

A number of practical applications that require registration involve partial or incomplete datasets, such as 3D reconstruction from Kinect measurements as in the case of KinectFusion [15]. In order to test the performance of MSBP for such applications, we consider an example of Stanford Armadillo man [7] (see Fig. 7-7). 500 points are sampled from the CAD model. In each run of the algorithm, one point is picked from the point set at random and the selected point along with 250 of its nearest neighbors are removed from the point set. The rest of the points are then used for registration with the original CAD model.

We observe that in spite of the lack of complete point set information, MSBP is able to correctly register the points to the CAD model. Fig. 7-7 shows two arbitrary runs of the algorithm with different sets of points missing in each. In both the cases, the MSBP correctly registers the points to the CAD model as shown in Fig. 7-7(a)-(b).

7.4 Conclusion and Discussion

In this chapter, we developed the multiple start branch and prune filtering algorithm (MSBP), a Kalman filter based method for nonconvex optimization. We show that using multiple initial states along with branching, updating and pruning, allows us to efficiently search for the optimal solution(s) in the domain of the search space without prematurely converging to a locally optimal solution. MSBP requires tuning of three parameters, the intuition behind which has been described and empirically verified with several examples. We show that the standard multi-hypothesis filter is a computationally less efficient, special case of the MSBP. With an example of point registration, MSBP is also compared with popular methods for nonconvex optimization and is found to estimate the optimal solutions accurately with a higher success rate especially when: 1) the objective function is available in an analytical form, 2) each function evaluation is expensive, 3) there are multiple global/local minima, and 4) the parameter space is relatively low dimensional (< 20).

Future work will involve an intermediate step to cluster the updated child states

instead of using an ϵ threshold. By using an information filter instead of a Kalman filter, the expensive matrix inversion operation step in the state update can be avoided. This would allow us to extend the MSBP for problems involving high dimensional parameter spaces. Validating the effectiveness of MSBP on a variety of nonconvex problems with different functional complexities, different number of parameters, and studying parameter sensitivity, will be a part of our future publication.

7.5 Contribution

The contribution from this chapter is:

1. Development of a filtering approach for nonconvex optimization. The approach can be used for probabilistic pose estimation with unknown data association.

2. The approach uses uncertainty for perturbation and innovation for pruning.

7.6 Published Work

Material from this chapter has appeared in the following publication

1. R Arun Srivatsan and Howie Choset, "Multiple Start Branch and Prune Filtering Algorithm for Nonconvex Optimization, in proceedings of the Workshop on the Algorithmic Fundamentals of Robotics, San Francisco, USA, December 2016.

Chapter 8

Sparse Point Registration

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 measurements of the object's surface, needs to be estimated (see Fig. 8-1). 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 [11, 12, 22, 5, 23]. 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 [222, 5, 23], or uses *a priori* knowledge of anatomical landmarks to hand-pick a small number of probing locations [70, 71, 16, 223]. Simon *et. al.* [70] find the best points to probe my minimizing a stiffness metric. Given a point to probe on the model, they use prior knowledge of anatomical segments to find a point close to the desired point on the actual anatomy. The

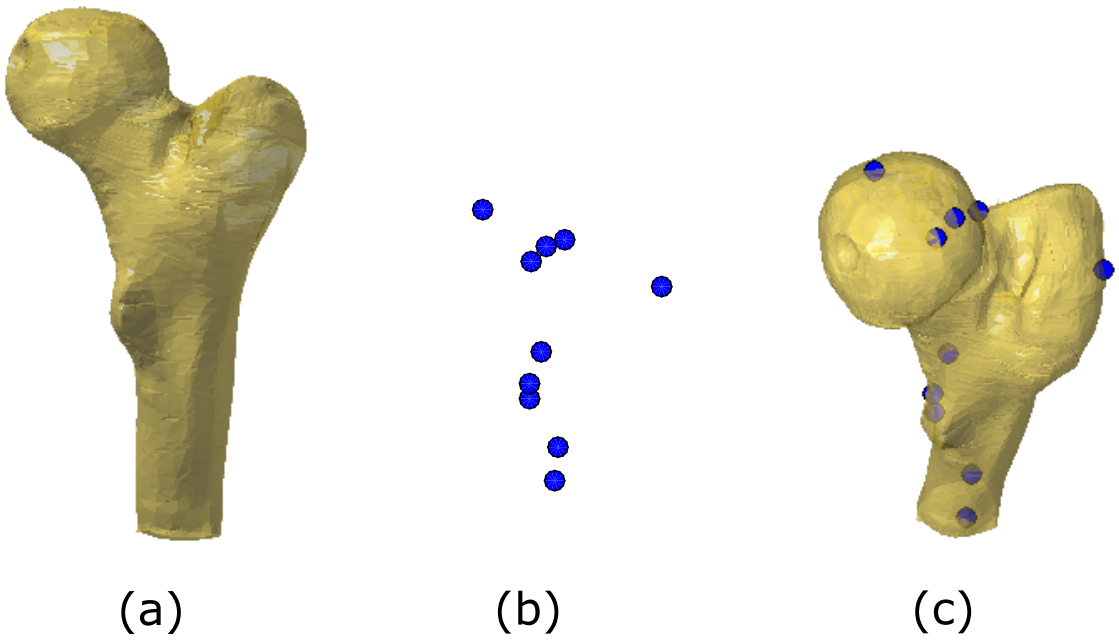


Figure 8-1: (a) Geometric model of the object (b) Point measurements in sensor frame (c) Point measurements registered to the geometric model

work of Ma *et. al.* [71, 16], extends upon the work of Simon *et. al.*, but they still require manually probing points on the right anatomical segments. The approach of Glozman *et. al.* [223] avoids choosing points based on the stiffness metric, but instead precomputes a small number of best points to probe by perturbing the model and computing which subset of points most affects the registration 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. In an attempt to keep the formulation general, in this work we do not assume any prior knowledge of anatomical segments and do not use the help of experts for any pre-registration.

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 poses obtained after perturbation 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 (ICP) [11], while the pSPR uses dual quaternion-based filter (DQF) (see Chapter 4) to estimate the pose. Note that this approach may appear similar to the stochastic ICP developed by Penney *et. al.* [224]. 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 Chapter 4 to process batches of point measurements.

In Sec. 8.2, 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. 8.2, a general guideline is provided on how to automatically probe the object to get a good spread of sparse points. Sec. 8.2 also shows the experimental results where a robot automatically probes several objects and accurately registers them to their geometric models.

8.1 Problem Formulation

8.1.1 Batch Dual Quaternion Filter

Dual quaternion-based filter (DQF) is a linear Kalman filtering based approach for online pose estimation, developed in Chapter 4. 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 uses pairs of measurements to update the registration estimate. Compared to other filtering based registration methods such as [6, 5, 49], the DQF is preferred because it is a truly linear filter without any approximations or linearizations, resulting in accurate estimates which are quickly obtained. The transformation $\boldsymbol{T} \in SE(3)$ is parameterized using the unit dual quaternion $\boldsymbol{x} = (\boldsymbol{q}, \boldsymbol{d})^T$, where $\boldsymbol{q} \in \mathbb{R}^4$ is a unit quaternion that parameterizes the rotation, $\boldsymbol{d}$ is the dual component of $\boldsymbol{x}$, $\boldsymbol{d} = \left((0, \boldsymbol{t})^T \odot \boldsymbol{q}\right)/2$ and $\odot$ is the quaternion multiplication operator, $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector.

In Chapter 7, we show that using a pair of measurements

$$\boldsymbol{H}\boldsymbol{q} = \boldsymbol{0}, \quad \boldsymbol{H} \in \mathbb{R}^{4\times 4} \tag{8.1}$$

$$\boldsymbol{H} = \begin{bmatrix} 0 & -(\boldsymbol{c}_1 - \boldsymbol{c}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2)^T \\ (\boldsymbol{c}_1 - \boldsymbol{c}_2 - \boldsymbol{b}_1 + \boldsymbol{b}_2) & (\boldsymbol{c}_1 - \boldsymbol{c}_2 + \boldsymbol{b}_1 - \boldsymbol{b}_2)^\times \end{bmatrix}, \tag{8.2}$$

$$\boldsymbol{t} = \frac{\boldsymbol{c}_1 - \boldsymbol{c}_2}{2} - Real\left(\boldsymbol{q} \odot \frac{\boldsymbol{b}_1 - \boldsymbol{b}_2}{2} \odot \boldsymbol{q}^*\right), \tag{8.3}$$

where $\boldsymbol{c}_i$ and $\boldsymbol{b}_i$ are points in the two reference frames that need to be registered, $\boldsymbol{q}^*$ 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:

$$\begin{aligned} \boldsymbol{q}_k &= \boldsymbol{q}_{k-1} - \boldsymbol{K}_k\boldsymbol{H}_k\boldsymbol{q}_{k-1}, \\ \boldsymbol{\Sigma}^{\boldsymbol{q}}_k &= (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{\Sigma}^{\boldsymbol{q}}_{k-1}, \quad \text{where,} \\ \boldsymbol{K}_k &= \boldsymbol{\Sigma}^{\boldsymbol{q}}_{k-1}\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{\Sigma}^{\boldsymbol{q}}_{k-1}\boldsymbol{H}_k^T + \boldsymbol{Q}_k)^{-1} \end{aligned}$$

where $\boldsymbol{\Sigma}^{\boldsymbol{q}}_k$ is the uncertainty in the quaternion $\boldsymbol{q}_k$ and $\boldsymbol{Q}_k$ is the pseudo-measurement uncertainty. The translation vector $\boldsymbol{t}_k$ is obtained from $\boldsymbol{q}_k$ using Eq. 8.3. For the sake of brevity, the derivation for $\boldsymbol{Q}_k = \boldsymbol{g}(\boldsymbol{\Sigma}^{\boldsymbol{q}}_k, \boldsymbol{\Sigma}^{\boldsymbol{b}_j}_k, \boldsymbol{\Sigma}^{\boldsymbol{c}_j}_k)$ and $\boldsymbol{\Sigma}^{\boldsymbol{t}}_k = \boldsymbol{f}(\boldsymbol{\Sigma}^{\boldsymbol{q}}_k, \boldsymbol{\Sigma}^{\boldsymbol{b}_j}_k, \boldsymbol{\Sigma}^{\boldsymbol{c}_j}_k)$ representing the uncertainty in the translation, are omitted here. The expressions along with their derivation can be obtained from Sec. 4.2.3.

The standard implementation of DQF requires ≈ 100 measurements for reliable

registration estimation. 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 bDQF. As shown in Sec. 8.2, bDQF requires fewer measurements for accurate estimation compared to the standard DQF.

We modify Eq. 8.1 as

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

where $\boldsymbol{H}_i \in \mathbb{R}^{4\times4}$ is as defined in Eq. 8.2 and $\alpha = \lfloor\left(\frac{m}{2}\right)\rfloor$. The update equations of the filter remain the same. The algorithm for bDQF is shown in Alg. 3.

Input:
$A = \{\boldsymbol{a}_i \in \mathbb{R}^3\}, i = 1, 2, ..., n$
$B = \{\boldsymbol{b}_j \in \mathbb{R}^3\}, j = 1, 2, ..., m$
Initial transformation: $\boldsymbol{q}_0 \in \mathbb{R}^4, \boldsymbol{t}_0 \in \mathbb{R}^3$,
Output:
$\boldsymbol{q} \in \mathbb{R}^4, \boldsymbol{t} \in \mathbb{R}^3$ that aligns A and B
$\boldsymbol{\Sigma}^q \in \mathbb{R}^{4\times4}, \boldsymbol{\Sigma}^t \in \mathbb{R}^{3\times3}$
Initialize: $k = 1$
while *not converged* **do**
  Correspondence:
  $\boldsymbol{T}_{k-1}(\boldsymbol{b}_j) = \boldsymbol{t}_{k-1} + Real(\boldsymbol{q}_{k-1} \odot \boldsymbol{b}_j \odot \boldsymbol{q}^*_{k-1})$
  $\boldsymbol{c}_j = FindClosestPoint(\boldsymbol{T}_{k-1}(\boldsymbol{b}_j))$, $\boldsymbol{c}_j \in A$,
  State Update:
  $\boldsymbol{q}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k)\boldsymbol{q}_{k-1}$
  $\boldsymbol{\Sigma}^{\boldsymbol{q}}_k = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{G}_k)\boldsymbol{\Sigma}^{\boldsymbol{q}}_{k-1}$
  $\boldsymbol{t}_k = \frac{1}{m}\left(\sum_{j=1}^m \boldsymbol{c}_j - Real\left(\boldsymbol{q}_k \odot \sum_{j=1}^m \boldsymbol{b}_j \odot \boldsymbol{q}^*_k\right)\right)$
  $\boldsymbol{\Sigma}^{\boldsymbol{t}}_k = \boldsymbol{f}(\boldsymbol{\Sigma}^{\boldsymbol{q}}_k, \boldsymbol{\Sigma}^{\boldsymbol{b}_j}_k, \boldsymbol{\Sigma}^{\boldsymbol{c}_j}_k)$
  $k = k + 1$
end

Algorithm 3: Batch Dual quaternion filtering

8.1.2 Steps Involved

Fig. 8-2 shows the basic framework of our SPR approach. The various steps involved are as follows:

1. The algorithm 3 is initialized using an initial pose (see Fig. 8-2(b)).

2. The current best pose is perturbed and p perturbed poses are obtained. In Fig. 8-2, $p = 3$ is chosen. The amount of perturbation is reduced over the iterations. Refer Sec. 8.1.3 and Sec. 8.1.4 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:

$$O_j = \sum_{i=1}^{m} ||\tilde{\boldsymbol{T}}_j(\boldsymbol{b}_i) - \boldsymbol{c}_i||, \quad j = 1, \ldots, p,$$

   where $\tilde{\boldsymbol{T}}_j \in SE(3)$ and $\boldsymbol{c}_i \in A$ is the closest point in the A. In this step, we use an approximate geometric model A^* instead of A, to quickly evaluate the cost function. Depending on the format of the geometric model, several existing simplification techniques can be applied, such as [223, 225, 226]. For example, when working with a triangulated mesh model, a quadric mesh simplification can be used [227] as shown in Fig. 8-2(c).

4. The pose $\hat{\boldsymbol{T}} = \text{argmin}_{\tilde{\boldsymbol{T}}_j} O_j$, is chosen as the initial guess for a locally optimal pose estimation using ICP or bDQF. In Sec. 8.2 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.

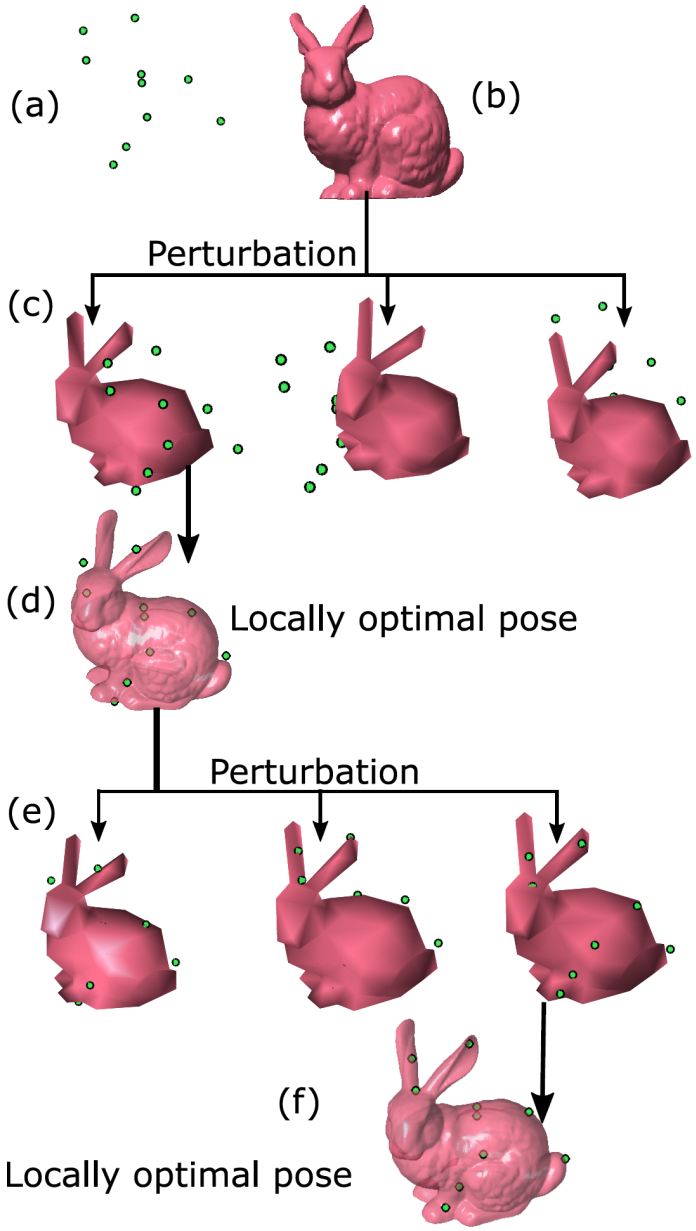


Figure 8-2: Figure shows the steps involved in an example of sparse point registration. In this example, two iterations of the algorithm are shown. The best pose estimate in each iteration is perturbed to obtain three pose particles. (a) Point measurements (b) Geometric model of the object (c) The geometric model in different perturbed poses. An approximate geometric model is used in this step. The number of triangle vertices in the original model is 259,896 and the number of vertices in the approximate model is 88. (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 best pose estimated from the previous iteration is perturbed to obtain three new poses. The perturbation in this step is lower than the previous iteration. (f) The locally optimal pose obtained after using ICP or bDQF. Note that the pose estimated in the second iteration provides an improvement over the previous iteration.

Input:
$A = \{\boldsymbol{a}_i \in \mathbb{R}^3\}, i = 1, 2, ..., n$
$B = \{\boldsymbol{b}_j \in \mathbb{R}^3\}, j = 1, 2, ..., m$
Initial transformation: $\boldsymbol{T}_0 \in se(3)$
Output: $\boldsymbol{T} \in SE(3)$ that aligns A and B
Initialize: $\boldsymbol{T} \leftarrow \boldsymbol{T}_0$, $k = 0$, $\epsilon = \inf$
while $k <$*MaxIterations OR* $\epsilon >$*Threshold* **do**
  Perturbation: $\tilde{\boldsymbol{T}}_j = \boldsymbol{T}_k + \mathcal{N}(0, \boldsymbol{\Sigma}_k)$, $j = 1, \ldots, p$
  Evaluate Cost Function: $O_j = \sum_{i=1}^{m} ||\tilde{\boldsymbol{T}}_j(\boldsymbol{b}_i) - \boldsymbol{c}_i||$
  Locally Optimal estimate:
  $\hat{\boldsymbol{T}}_k = \text{argmin}_{\tilde{\boldsymbol{T}}_j} O_j$
  $\boldsymbol{T}_k = \text{ICP}(A, B, \hat{\boldsymbol{T}}_k)$ or $\text{DQF}(A, B, \hat{\boldsymbol{T}}_k)$
  $\epsilon_k = \sum_{i=1}^{m} ||\boldsymbol{T}_k(\boldsymbol{b}_i) - \boldsymbol{c}_i||$
  if $\epsilon_k < \epsilon$ **then**
    $\boldsymbol{T} = \boldsymbol{T}_k$, $\epsilon = \epsilon_k$
  end
  $k = k + 1$
end

Algorithm 4: Sparse Point Registration

8.1.3 Deterministic Sparse Point Registration (dSPR)

In the dSPR, ICP as described in Sec. 2.1 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 chapter 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 chapter 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 10mm in translation. 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 the two point sets is lower than a set threshold. In this chapter, we set the maximum number of iterations to be 30 and the RMS error threshold to be 0.5mm. In addition the maximum number of iterations of each ICP step is set to 20.

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

8.1.4 Probabilistic Sparse Point Registration (pSPR)

In the pSPR, bDQF as described in Sec. 8.1.1 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 (refer to the UKF for a discussion on the sigma points [181]). 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 chapter, we set the maximum number of iterations to be 12 and the RMS error threshold to be 0.5mm. The number of iterations in each bDQF step is set to 50.

8.2 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 unscented particle filter (UPF) developed by Ma *et. al.* [16]. 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 [1].

8.2.1 Minimum Number of Points Required

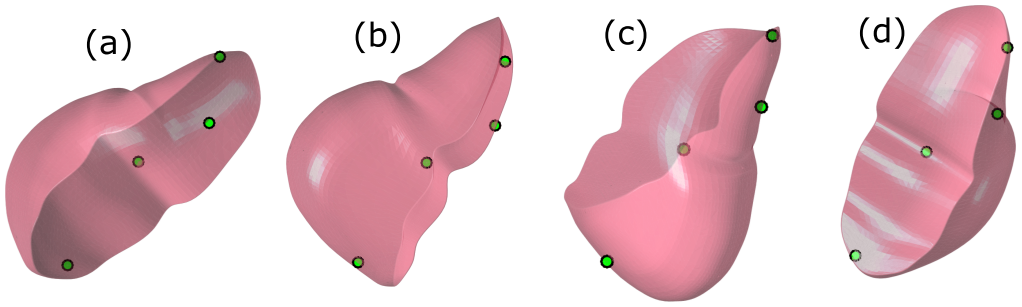


Figure 8-3: Four different poses of the liver contain the same set of four green points.

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 [4]. 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. 8-3). 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.* [70] (later extended by Ellis *et. al.* [72]) 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 minimum 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

[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.

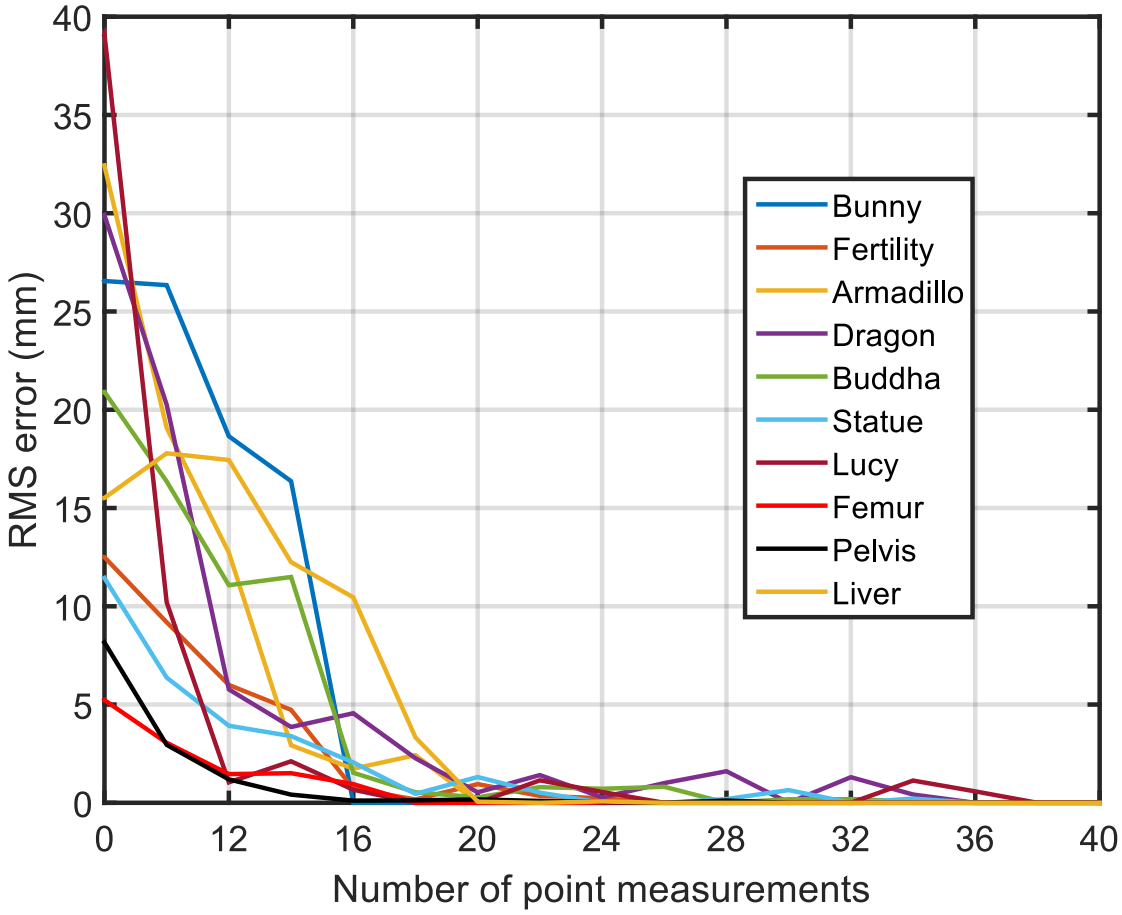


Figure 8-4: Plot of RMS error vs number of points used for registration, when using dSPR. For each integer element on the X axis, mean error is computed over 100 experiments. Most of the shapes considered need $\approx$ 20 measurements for accurate registration.

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 [7], Fertility obtained from the AIM shape repository [228], 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.

8.2.2 Robustness to Noise

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. 8-5 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. 8-5 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 8.1 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

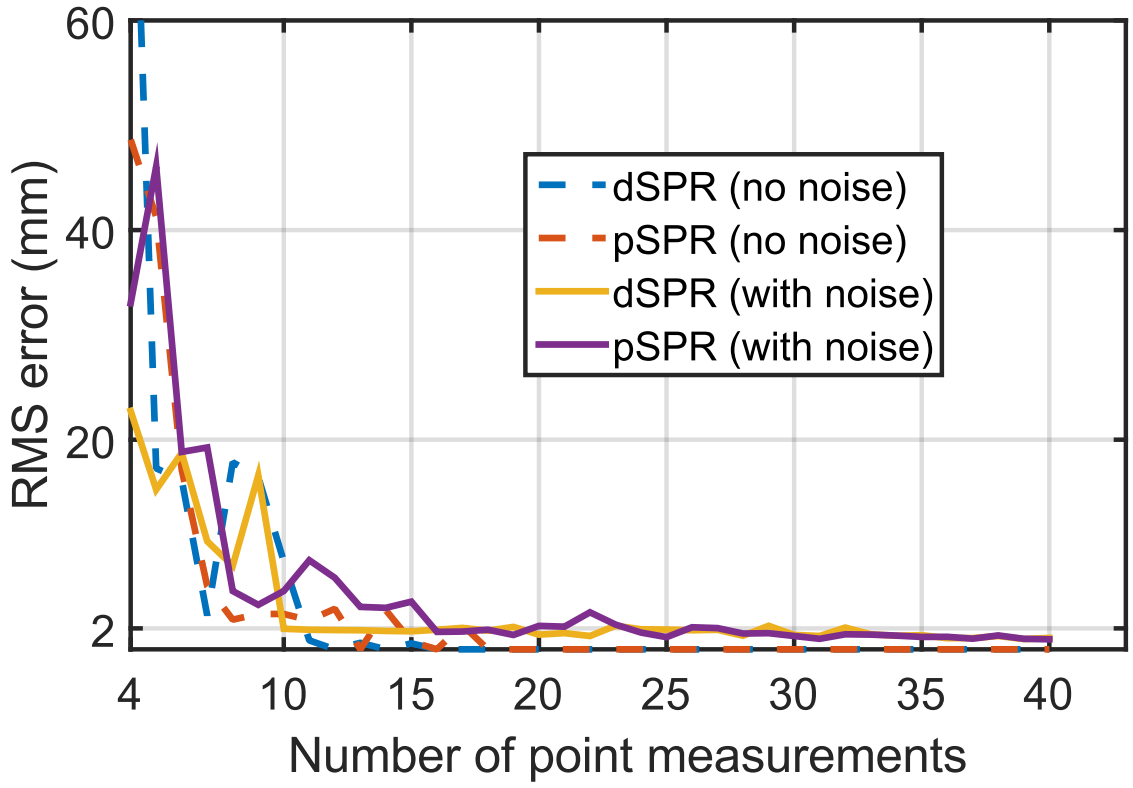


Figure 8-5: Plot of the RMS error vs number of measurements used for dSPR and pSPR, with a without noise in the measurements. In the absence of noise, dSPR takes 12 measurements and pSPR takes 18 measurements to converge to zero RMS error. In the presence of a uniform noise of 2mm, both pSPR and dSPR converge to an RMS error of $<$ 2mm after 20 measurements.

to iterate over $\approx$ 2000 pose particles (other researchers have also noticed the high computation time of UPF [5]).

In this work, we do not present comparisons with other popular registration methods such as generalized ICP [22], UKF-based registration [5], DQF [50], iterative most likely point (IMLP) [23], etc. as those methods are not designed to work with less than 100 measurements and hence the comparison would be unfair.

Table 8.1: 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	1.00	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 $\approx$ 20 iterations, while each bDQF evaluation in pSPR internally takes $\approx$ 50 iterations for convergence to a local minima. The benefit of using pSPR however, lies in the fewer parameters that require tuning.

8.2.3 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. 8-6(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. 8-6(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 submodularity-based touch of Javdani *et. al.* [229], or manually by a teleoperator as in our prior work [230].

8.3 Robot Experiments

In order to test our approach to SPR with real data, an experimental setup as shown in Fig. 8-7(a) is used. The setup consists of a 6-DOF robot Foxbot® equipped with

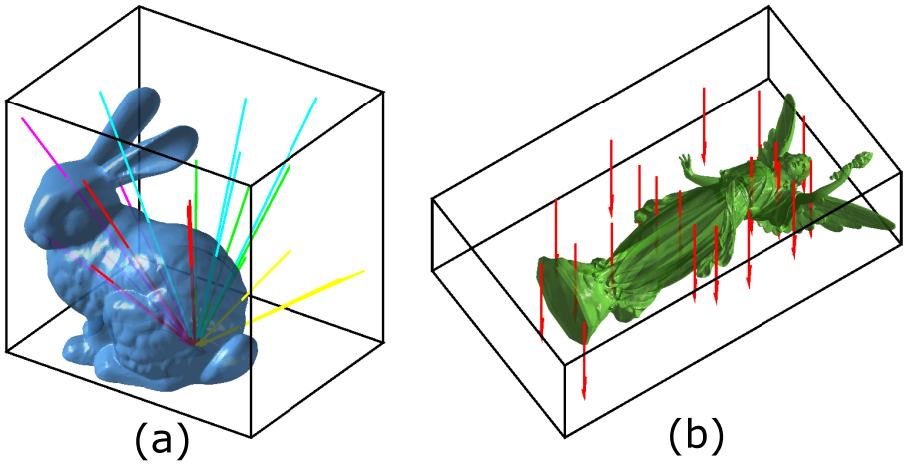


Figure 8-6: A cuboid is selected in the workspace of the robot that conservatively estimates the location of the object. (a) Different probing paths for the robot are selected such that the probed points are spread across the surface of the object. The colors of the path show the face of the cuboid that 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.

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. 8.2.3. 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. 6-10, we collect 18, 20 and 20 points respectively for the femur, pelvis and bunny respectively. The following video shows the experiment in progress: https://youtu.be/xnkGRFBDE0Q

Table 8.2: 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

The blue circles in Fig. 8-7(b)-(d) show the initial guess for the location of the point measurements and the red circles show the location as estimated by dSPR.

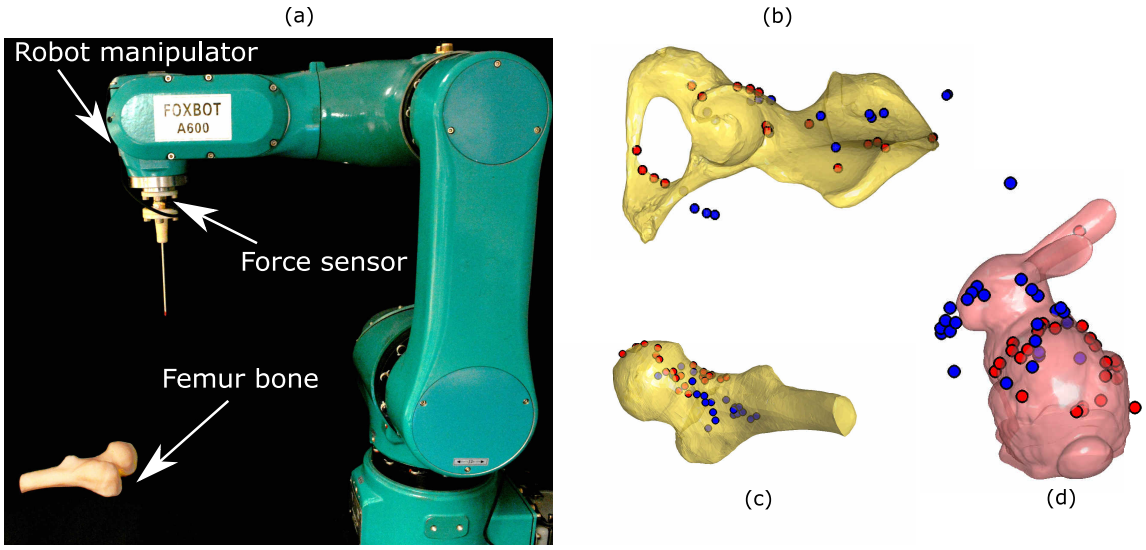


Figure 8-7: (a) Experimental set up consists of a robot manipulator equipped with a force sensor. The object that is to be registered is clamped and held in place. (b)-(d) Blue circles represent the initial location of the point measurements and red circles represent the registered location of the points. Pelvis bone, femur bone and bunny are probed at 18, 20 and 20 points respectively.

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 8.2. 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. 6-10).

8.4 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, 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 complementary model update (see Chapter 3) 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.

8.5 Contribution

The contributions from this chapter include:

1. A probabilistic approach that estimates registration using a small number of sparse point measurements.

2. A deterministic variant was also developed that can estimate registration with sparse point measurements. This variant however, requires tuning a number of parameters.

8.6 Published Work

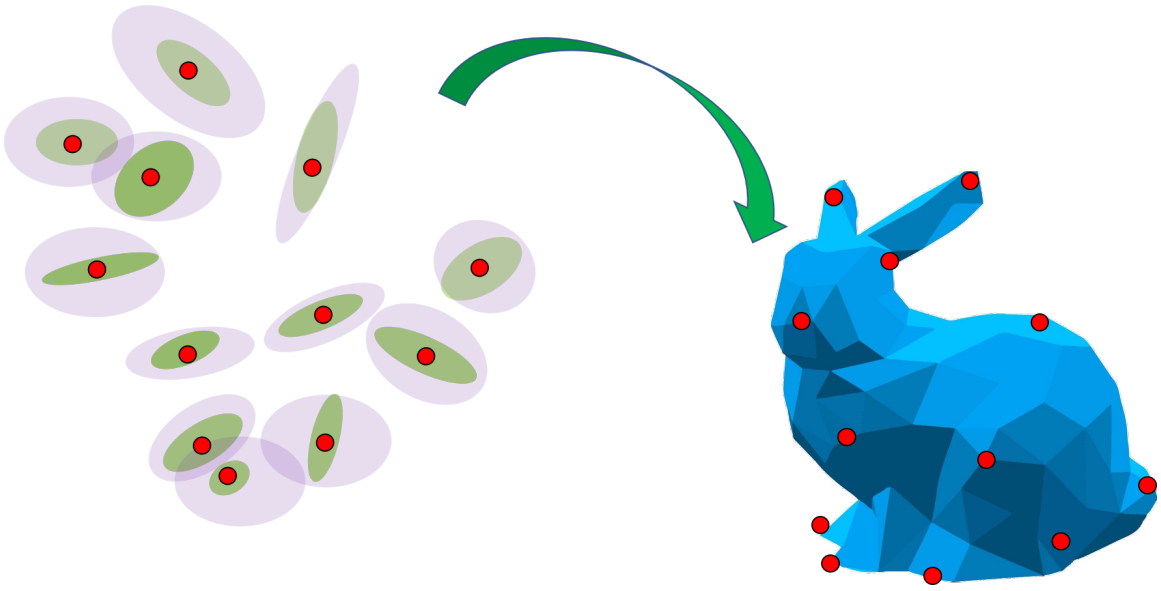
Material from this chapter has appeared in the following publications

1. R Arun Srivatsan, Prasad Vagdargi and Howie Choset,"Sparse Point Registration, in proceedings of the International Symposium on Robotics Research, 2017.

2. R Arun Srivatsan, Prasad Vagdargi, Nicolas Zevallos and Howie 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.

Part IV

Generalizing prior methods into a common framework



Chapter 9

Generalized Framework for Probabilistic Pose Estimation

In this chapter we focus on registering a point cloud obtained in sensor frame to a point cloud or a surface model defined in model frame (see Fig. 9-1). Our framework generalizes a number of popular registration approaches and provides registration estimates that are equivalent to or more accurate than existing methods. By restricting our scope of study to point measurements, and assuming uncertainties in both position measurements and prior pose parameters, we derive an objective function to formulate registration as an optimization problem. Minimizing the objective function guarantees global optimality. We also derive a locally optimal variant for registration which is based on an expectation maximization (EM) framework. This variant is computationally faster than the globally optimal variant, although it only guarantees convergence to a local optimum. Instead of burdening a user with having to try a large number of existing methods in literature and then choose the one that best works for their application, we provide the user a framework that considers point-set registration at its most generic form with a recipe to obtain optimal registration, for their application.

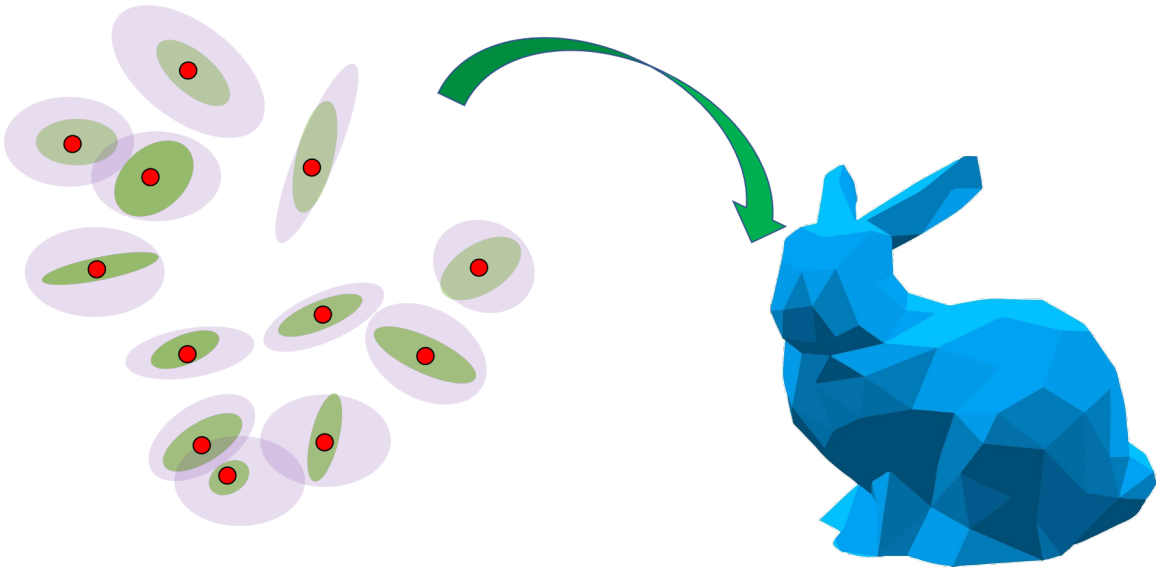


Figure 9-1: Red points on the left need to be registered to a model of Stanford bunny on the right. The ellipses in green show the uncertainty in the measurements, while purple ellipses show the uncertainty due to both measurement noise and prior registration estimate.

9.1 Related Work

Several approaches to point set registration have been developed over the last few decades (for a review refer to [86]). Registration is always a nonconvex problem irrespective of our knowledge of the correspondence between the points in the two reference frames. Despite this, there are methods in literature that guarantee global optimality when correspondence in known. For example if the uncertainty in the point measurements is isotropic, Horn's method [4], Arun's method [17], the quaternion-based approach of Faugeras and Hebert [60] and the dual quaternion-based approach of Walker *et. al.* [61] provide theoretical guarantees on the globally optimal registration. If the uncertainty in point measurements is anisotropic, then methods such as generalized ICP (G-ICP) [22], generalized total least squares ICP (GTLS-ICP) [21] and IMLP [23] can guarantee optimal registration. And if the uncertainty in point measurements is anisotropic and there exists a prior pse uncertainty, then the Bingham distribution-based filter (BF) described in Chapter 6 provides optimal registration estimates.

However, in many practical applications the correspondence is unknown. The iterative closest point (ICP) algorithm introduced by Besl and McKay [11] is a widely popular algorithm that alternates between finding the best correspondence and the optimal transformation given the correspondence. Many variants of the ICP have ince been proposed (see [231] for a review). Some of these variants use surface normals [232], curvature [233], color-intensity [234], histogram of features [74], etc., in addition to point measurements. Others are robust to outliers [66, 235, 236], use alternate means to find correspondence [237], or use global optimizers [27, 238, 18, 20].

Probabilistic registration approaches have been developed to better handle the noise in the measurements. For example, the works of Wells *et. al.* [239], Chui *et. al.* [240], Granger *et. al.* [241, 85](expectation maximization ICP (EM-ICP)), use an EM framework, wherein each point in one set, is mapped to every point in the other set with a weight assigned for the correspondence (this is referred to as *soft matching*). An objective function which is composed using these weights is then minimized to obtain the optimal registration. This is in stark contrast to methods such as ICP where the objective function uses a single closest point in the model frame that is estimated for each sensor point (this kind of matching is referred to as *hard matching*). Algorithms incorporating soft matching tend to achieve higher accuracy and have wider basins of convergence towards the optimum compared to methods that use hard matching [85, 116].

While many of the above methods assume equal isotropic noise in all point measurements, anisotropic noise is more frequently encountered in realworld measurements such as from range sensors [22], tomographic imaging [242], stereo imaging [243], etc. The generalized total least squares ICP (GTLS-ICP) [21], and generalized ICP (G-ICP) [22] account for anisotropic noise in the measured data points. However, they use a closest point matching, in the correspondence phase. The anisotropic ICP (A-ICP) [244] extends the prior work by using a Mahalanobis distance metric for the point correspondence, that is a function of the sensor noise. The IMLP further improves upon the A-ICP by using a *most likely matching criteria*[1], instead of the

[1]The most likely matching criteria first finds the probability of a sensor point to be a potential

Table 9.1: Classification of various registration methods

Registering point cloud to point cloud			
Method	Noise	MLE or MAP	Correspondence
EM-ICP [239, 241]	Equal, Isotropic	MLE	Probabilistic soft
GTLS-ICP [21]	Anisotropic	MLE	Closest point
G-ICP [22]	Anisotropic	MLE	Closest point
A-ICP [244]	Anisotropic	MLE	Mahalanobis distance
IMLP [23]	Anisotropic	MLE	Most likely
Bingham filter [245]	Anisotropic	MAP	Closest point
Our approach	Anisotropic	MAP	Probabilistic soft
Registering point cloud to surface model			
ICP [11]	Equal, Isotropic	MLE	Closest point
IMLP [23]	Anisotropic	MLE	Most likely
EKF [6]	Anisotropic	MAP	Closest point
IEKF [64]	Anisotropic	MAP	Closest point
UKF [5]	Anisotropic	MAP	Closest point
UKF [49]	Anisotropic	MAP	Closest point
Bingham filter [245]	Anisotropic	MAP	Closest point
Our approach	Anisotropic	MAP	Probabilistic soft

Mahalanobis distance [23].

All the methods described thus far do not use prior pose information, and can be thought of as using maximum likelihood estimate (MLE) to find the optimal transformation. However, in applications where reliable prior information is available, maximum *a posteriori* (MAP) estimate provides more accurate registration. For example, methods such as ICP only use point locations (red points in Fig. 9-1), while methods such as IMLP use point locations and sensor noise (red points and green ellipses in Fig. 9-1). In a MAP estimate approach, point locations, sensor noise and uncertainty in prior pose are all used (red points and purple ellipses in Fig. 9-1). The registration methods that use MAP estimate often utilize Bayesian filters, such as EKF [6], IEKF [64], and UKF [5, 49]. While these methods resort to linearization or some other form of approximation, the Bingham filter-based pose estimation (see

match to each model point. The model point that is most probable is then used in the objective function that is optimized to obtain the best registration estimate. If the sensor noise was isotropic, then the Mahalanobis distance of A-ICP, the most likely matching criteria of IMLP and the closest point correspondence of ICP become identical.

Chapter 6) avoids approximations by using a truly linear update model.

In this chapter we introduce a probabilistic framework which is most similar to IMLP, but can perform both MLE and MAP estimation. The framework is also similar to the Bingham filter, but it uses probabilistic correspondence instead of closest point matching. In addition, we unify disparate point set registration approaches into a single probabilistic framework and reasons their application in terms of – MLE or MAP estimate, as shown in Table 9.1. We show that our framework formulates an optimization problem for the most general case of registration and by appropriately setting the functional parameters such as sensor noise, and prior pose distribution, etc., the user is guaranteed to obtain the optimal registration for their application without having to pick and choose from a large number of methods in literature.

9.2 Problem Formulation

We discuss the generalized framework for two broad cases of registration –

1. Registering a point cloud to another point cloud (an example is structure from motion [22]),

2. Registering a point cloud to a surface model (for example localizing surgical tool [16]).

9.2.1 Case 1: Registering point cloud to point cloud

Let $\mathcal{M} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{n_\mathcal{M}}\}$ be the set of points in the model frame and $\mathcal{S} = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{n_\mathcal{S}}\}$ be the set of points in the sensor frame, such that $\boldsymbol{m}_i, \boldsymbol{s}_i \in \mathbb{R}^3$. Let $\boldsymbol{x}$ be the set of parameters that represent the transformation given by rotation $\boldsymbol{R} \in SO(3)$, and translation $\boldsymbol{t} \in \mathbb{R}^3$. Then the optimal registration parameters are obtained using a maximum *a posteriori* (MAP) estimate as shown,

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}}\, P(\boldsymbol{x}|\mathcal{S}, \mathcal{M}). \tag{9.1}$$

Table 9.2: Notations

Symbol	Description
$\boldsymbol{I}_{n\times n}$	$n \times n$ identity matrix
$\widetilde{\boldsymbol{m}}$	$(0, \boldsymbol{m}^T)^T \forall \boldsymbol{m} \in \mathbb{R}^3$
$\widetilde{\boldsymbol{M}}$	$\begin{bmatrix} 0 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix} \in \mathbb{R}^{4\times4} \forall \boldsymbol{M} \in \mathbb{R}^{3\times3}$
$\widehat{\boldsymbol{M}}$	$\begin{bmatrix} 1 & \boldsymbol{0}^T \\ \boldsymbol{0} & \boldsymbol{M} \end{bmatrix} \in \mathbb{R}^{4\times4} \forall \boldsymbol{M} \in \mathbb{R}^{3\times3}$
$\overline{\boldsymbol{M}}$	Last three columns of $\boldsymbol{M} \in \mathbb{R}^{4\times4}$

Upon invoking Bayes rule,

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \underbrace{P(\boldsymbol{x})}_{\text{prior}} \underbrace{P(\mathcal{S}|\mathcal{M}, \boldsymbol{x})}_{\text{likelihood}}. \tag{9.2}$$

Let $W \in \mathbb{R}^{n_\mathcal{S} \times n_\mathcal{M}}$ be the matching matrix. Assuming there are no outliers, each sensor point is mapped to a unique point on the model and hence we define W to take binary values with each row having only one entry whose value is 1 (Sec. 9.2.3 describes how to deal with outliers). We rewrite Eq. 9.2 by bringing in an explicit dependence on W

$$\begin{aligned} P(\mathcal{S}|\mathcal{M}, \boldsymbol{x}) &= \sum_{\{W\}} P(\mathcal{S}, W|\mathcal{M}, \boldsymbol{x}), \\ &= \sum_{\{W\}} P(W|\mathcal{M}, \boldsymbol{x}) P(\mathcal{S}|W, \mathcal{M}, \boldsymbol{x}). \end{aligned} \tag{9.3}$$

If we consider a random matching matrix $\boldsymbol{W}$, then each possible matching matrix W has a probability $P(\boldsymbol{W} = W)$. Since W_{ij} (the ij^{th} element of W) takes binary values, the expected value of W_{ij} is obtained as follows

$$\begin{aligned} E(\boldsymbol{W}_{ij}) &= \sum_{W_{ij}=0,1} W_{ij} P(\boldsymbol{W}_{ij} = W_{ij}) \\ &= P(\boldsymbol{W}_{ij} = 1). \end{aligned}$$

The probability of obtaining W, is

$$P(\boldsymbol{W} = W) = \prod_{i=1}^{n_{\mathcal{S}}} \prod_{j=1}^{n_{\mathcal{M}}} (\underbrace{P(\boldsymbol{W}_{ij} = 1)}_{\phi_{ij}})^{W_{ij}}, \tag{9.4}$$

since W_{ij} is binary. It is common practice to assume that each model point is equally likely given no prior information about sensed points, i.e., $\phi_{ij} = \frac{1}{n_{\mathcal{M}}}$. Thus we have

$$P(\boldsymbol{W} = W|\mathcal{M}, \boldsymbol{x}) = \prod_{i=1}^{n_{\mathcal{S}}} \prod_{j=1}^{n_{\mathcal{M}}} \left(\frac{1}{n_{\mathcal{M}}}\right)^{W_{ij}}. \tag{9.5}$$

The likelihood of $\mathcal{S}$ given W is

$$P(\mathcal{S}|W, \mathcal{M}, \boldsymbol{x}) = \prod_{i,j} (P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x}))^{W_{ij}}. \tag{9.6}$$

Substituting Eq. 9.5, 9.6, in Eq. 9.3,

$$\begin{aligned} P(\mathcal{S}|\mathcal{M}, \boldsymbol{x}) &= \sum_{\{W\}} \prod_{i,j} \left(\frac{1}{n_{\mathcal{M}}} P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right)^{W_{ij}}, \\ &= \prod_{i=1}^{n_{\mathcal{S}}} \left(\sum_{j=1}^{n_{\mathcal{M}}} \frac{1}{n_{\mathcal{M}}} P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right). \end{aligned} \tag{9.7}$$

Substituting Eq. 9.7 in Eq. 9.2 and taking the logarithm gives

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \log(P(\boldsymbol{x})) + \sum_{i=1}^{n_{\mathcal{S}}} \log \left(\sum_{j=1}^{n_{\mathcal{M}}} \frac{1}{n_{\mathcal{M}}} P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right). \tag{9.8}$$

There is no known closed form solution for $\boldsymbol{x}$ in Eq. 9.8, and to the best of our knowledge there is no method in literature to directly minimize this criterion robustly [241, 85]. Global optimizers such as GA, PSO, branch and bound, etc. can be used to obtain optimal pose from Eq 9.8.

Since it is evident from literature than using global optimizers results in larger computation times [], we also develop a faster albeit locally optimal approach for registration. This formulation is inspired by Wells [239], who us an EM approach to

find $\boldsymbol{x}^*$. From Eq. 9.2,

$$\begin{aligned}\boldsymbol{x}^* &= \underset{\boldsymbol{x}}{\operatorname{argmax}} \underbrace{P(\boldsymbol{x})\frac{P(\mathcal{S}, W|\mathcal{M}, \boldsymbol{x})}{P(W|\mathcal{S}, \mathcal{M}, \boldsymbol{x})}}_{P(\boldsymbol{x}|\mathcal{S},\mathcal{M})}, \\ &= \underset{\boldsymbol{x}}{\operatorname{argmin}} \underbrace{-\log\left(P(\boldsymbol{x}|\mathcal{S}, \mathcal{M})\right)}_{O(\boldsymbol{x})}. \end{aligned} \tag{9.9}$$

Since $O(\boldsymbol{x})$ is independent of W, the expectation with respect to $\boldsymbol{W}$ will remain the same, i.e., $O(\boldsymbol{x}) = E_{\boldsymbol{W}}(O(\boldsymbol{x}))$. From Eq. 9.2 and Eq. 9.9 we have

$$O(\boldsymbol{x}) = -E_{\boldsymbol{W}}(\log(P(\boldsymbol{x}))) - E_{\boldsymbol{W}}(\log(P(\mathcal{S}, W|\mathcal{M}, \boldsymbol{x}))) + E_{\boldsymbol{W}}(\log(P(W|\mathcal{S}, \mathcal{M}, \boldsymbol{x}))).$$

Following the approach of [85], we add an explicit dependence on matching matrix using Kullback-Leibler (KL) divergence between $\boldsymbol{W}$ and the *a posteriori* likelihood of matches $P(W_{\boldsymbol{x}} = W) = P(W|\mathcal{S}, \mathcal{M}, \boldsymbol{x})$,

$$\begin{aligned} O(\boldsymbol{x}, \boldsymbol{W}) &= O(\boldsymbol{x}) + E_{\boldsymbol{W}}\left(\log\left(\frac{P(W)}{P(W|\mathcal{S}, \mathcal{M}, \boldsymbol{x})}\right)\right), \\ &= E_{\boldsymbol{W}}\left(\log(P(W))\right) - E_{\boldsymbol{W}}\left(\log(P(\boldsymbol{x}))\right) - E_{\boldsymbol{W}}\left(\log(P(\mathcal{S}, W|\mathcal{M}, \boldsymbol{x}))\right). \end{aligned} \tag{9.10}$$

The distance is null when $W_{\boldsymbol{x}} = W$, hence we have $O(\boldsymbol{x}) = \min_{\boldsymbol{W}} O(\boldsymbol{x}, \boldsymbol{W}) = O(\boldsymbol{x}, W_{\boldsymbol{x}})$.

Given the current best estimate of $\boldsymbol{x}$, we first find the expected value of W,

$$\begin{aligned} P(W_{\boldsymbol{x}} = W) &= \frac{P(\mathcal{S}, W|\mathcal{M}, \boldsymbol{x})}{P(\mathcal{S}|\mathcal{M}, \boldsymbol{x})} \\ &= \prod_{i=1}^{n_{\mathcal{S}}}\prod_{j=1}^{n_{\mathcal{M}}}\left(\frac{\phi_{ij}P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})}{\sum_{k=1}^{n_{\mathcal{M}}}\phi_{ik}P(\boldsymbol{s}_i|\boldsymbol{m}_k, \boldsymbol{x})}\right)^{W_{ij}}. \end{aligned}$$

Comparing with Eq. 9.4, we have

$$E\left((W_{\boldsymbol{x}})_{ij}\right) = \left(\frac{\phi_{ij}P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})}{\sum_{k=1}^{n_{\mathcal{M}}}\phi_{ik}P(\boldsymbol{s}_i|\boldsymbol{m}_k, \boldsymbol{x})}\right). \tag{9.11}$$

Following that, we use $W = W_{\boldsymbol{x}}$ to minimize Eq. 9.10 and obtain the best estimate

of $\boldsymbol{x}$. Note that only the last two terms in Eq. 9.10 are a function of $\boldsymbol{x}$, thus $\boldsymbol{x}_k = \text{argmin}_{\boldsymbol{x}} O(\boldsymbol{x}, W_{\boldsymbol{x}})$ where

$$O(\boldsymbol{x}, W) = -\log(P(\boldsymbol{x})) - \sum_{i,j} \underbrace{E(W_{ij})}_{w_{ij}} \log\left(P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right), \tag{9.12}$$

where $\boldsymbol{x}_k$ is the estimate of $\boldsymbol{x}$ at the k^{th} iteration. By alternating between expectation (Eq. 9.11) and minimization (Eq. 9.12), we are guaranteed to obtain a locally optimal solution for $\boldsymbol{x}$.

If there is no prior information about the distribution of the registration parameters, then the problem becomes a MLE instead of MAP. This implies, instead of Eq. 9.2, we would use the following

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\text{argmax}} \underbrace{P(\mathcal{S}|\mathcal{M}, \boldsymbol{x})}_{\text{likelihood}}. \tag{9.13}$$

Following the same derivation as for MAP, we would obtain

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\text{argmax}} \sum_{i=1}^{n_{\mathcal{S}}} \log\left(\sum_{j=1}^{n_{\mathcal{M}}} \frac{1}{n_{\mathcal{M}}} P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right). \tag{9.14}$$

Where $\boldsymbol{x}^*$ obtained from Eq. 9.14 is guaranteed to provide globally optimal registration. Similarly for the locally optimal variant, Eq. 9.12 can be rewritten as

$$O(\boldsymbol{x}, W) = -\sum_{i,j} \underbrace{E(W_{ij})}_{w_{ij}} \log\left(P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})\right). \tag{9.15}$$

By alternating between Eq. 9.11 and Eq. 9.15, we are guaranteed to obtain a locally optimal solution for $\boldsymbol{x}$ when no prior distribution for the registration parameters is available.

9.2.2 Case 2: Registering point cloud to a surface model

When registering two point clouds, a matching matrix W can be used to determine the correspondence, as shown earlier. However, such an approach does not scale well, when a point cloud needs to be registered to a surface. This is because the surface has infinite points lying on it and it is computationally intractable to find the match probability (as in Eq. 9.11) for each point on the model. Surface models are often discretized using parameters that vary in a continuous domain or by meshes composed of vertices, edges and faces (the faces are often polygonal in shape). There are prior works which sample finite number of points on the surface and use the matching matrix approach for registration [85]. Such an approach has been shown to be error prone and only approximate [85].

Irrespective of the discretization used, in this work we assume that each sensed point is homologous to a single point on the model. We define the optimization as

$$\begin{aligned}(\boldsymbol{x}^*, \overline{\boldsymbol{m}^*}) &= \underset{\boldsymbol{x}, \overline{\boldsymbol{m}}}{\operatorname{argmin}}\, O(\boldsymbol{x}, \overline{\boldsymbol{m}}), \quad \text{where} \\ O(\boldsymbol{x}, \overline{\boldsymbol{m}}) &= -\log\left(P(\boldsymbol{x})\right) - \sum_{i=1}^{n_{\mathcal{S}}} \log\left(P(\boldsymbol{s}_i|\overline{\boldsymbol{m}}, \boldsymbol{x})\right), \end{aligned} \tag{9.16}$$

where $\boldsymbol{x}^*$ is the optimal registration parameters, $\overline{\boldsymbol{m}^*}$ is the set of optimal correspondences in the model frame, $\overline{\boldsymbol{m}} = \{\boldsymbol{m}_1, \ldots, \boldsymbol{m}_{n_{\mathcal{S}}}\} \in \mathcal{M}$ are the points on the model that correspond to $\boldsymbol{s}_i$. However, we do not have *a priori* knowledge of the matches and hence we alternate between finding correspondence ($\overline{\boldsymbol{m}}$) and minimizing an objection function over $\boldsymbol{x}$.

The corresponding model points $\overline{\boldsymbol{m}}_{\boldsymbol{x}_k}$, given the current best estimate $\boldsymbol{x}_k$ is

$$\begin{aligned}\overline{\boldsymbol{m}}_{\boldsymbol{x}_k} &= \underset{\overline{\boldsymbol{m}}}{\operatorname{argmin}}\, O(\boldsymbol{x}_k, \overline{\boldsymbol{m}}), \\ \Rightarrow \boldsymbol{m}_i &= \underset{\boldsymbol{m} \in \mathcal{M}}{\operatorname{argmin}} - \log(P(\boldsymbol{s}_i|\boldsymbol{m}, \boldsymbol{x})). \end{aligned} \tag{9.17}$$

Keeping the updated matches $\overline{\boldsymbol{m}}_{\boldsymbol{x}_k}$ fixed, we find the best estimate of $\boldsymbol{x}$

$$\boldsymbol{x}_{k+1} = \underset{\boldsymbol{x}}{\operatorname{argmin}}\, O(\boldsymbol{x}, \overline{\boldsymbol{m}}_{\boldsymbol{x}_k}). \tag{9.18}$$

The iterative optimization setup ensures that $O(\boldsymbol{x}_k, \overline{\boldsymbol{m}}_{\boldsymbol{x}_{k-1}}) \geq O(\boldsymbol{x}_k, \overline{\boldsymbol{m}}_{\boldsymbol{x}_k}) \geq O(\boldsymbol{x}_{k+1}, \overline{\boldsymbol{m}}_{\boldsymbol{x}_k})$, i.e., with each iteration the function value decreases and thus after sufficient iterations we are guaranteed to obtain a locally optimal solution for both $\boldsymbol{x}$ and $\overline{\boldsymbol{m}}$.

If there is no prior information about the distribution of the registration parameters, then the problem becomes a MLE as discussed in the previous section. Instead of Eq. 9.16, we would use the following

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmin}} \underbrace{-\sum_{i=1}^{n_{\mathcal{S}}} \log\left(P(\boldsymbol{s}_i|\overline{\boldsymbol{m}}, \boldsymbol{x})\right)}_{O(\boldsymbol{x}, \overline{\boldsymbol{m}})}. \tag{9.19}$$

9.2.3 Outlier Rejection

There are number of standard procedures to handle outliers when performing registration [235, 23, 236, 27]. The choices that we found to be best suited for a number of applications are

- Distance thresholding: A chi-square test has been widely used and it is a simple way to detect outliers [5, 23]. When evaluating the expected matching using Eq. 9.24 or finding the most likely matches using Eq. 9.33, the $\log(P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x}))$ term is a square Mahalanobis distance (if you ignore the normalization constant). Outliers are detected by comparing square Mahalanobis distance at each point measurement to the inverse cumulative density function of a chi-square distribution with 4 degrees of freedom evaluated at some probability. Outliers are simply ignored from the further calculations in that iteration.

- Percentage rejection: If there is some heuristic that says what percentage of data would be outliers, then that percentage of sensed points with the farthest distance from the model is ignored in each iteration [18]. The distance metric

to be used can be the same Mahalanobis as described in the previous method. This methods works fine if the percentage of outliers is not very high. Otherwise it is observed to get trapped in local minima quickly.

9.3 Optimization Setup

9.3.1 Case1:Registering point cloud to point cloud

In order to find the optimal registration parameters from Eq. 9.12, we first derive the form of $P(\boldsymbol{x})$, $P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})$ and w_{ij}. The form of $P(\boldsymbol{x})$ considered here is

$$P(\boldsymbol{x}) = \frac{\exp\left(\frac{1}{2}\boldsymbol{q}^T\boldsymbol{D}\boldsymbol{q} - \frac{1}{2}(\boldsymbol{t}-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{\mu})\right)}{N_{\boldsymbol{D}}N_{\boldsymbol{C}}}, \tag{9.20}$$

where $N_{\boldsymbol{D}}, N_{\boldsymbol{C}}$ are normalization constants. $\boldsymbol{x} = (\boldsymbol{q}^T, \boldsymbol{t}^T)^T$, where $\boldsymbol{q} \in \mathbb{R}^4$ is the unit quaternion parameterizing the rotation and $\boldsymbol{t} \in \mathbb{R}^3$ is the translation vector.

Let us define $\boldsymbol{d}_{ij} = \boldsymbol{m}_j - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t}$. Rewriting this equation in terms of unit quaternions [61],

$$\begin{aligned}\widetilde{\boldsymbol{d}}_{ij} =& \widetilde{\boldsymbol{m}}_j - \Omega_1(\boldsymbol{q})^T\Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{s}}_i - \widetilde{\boldsymbol{t}},\\ \Rightarrow \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{d}}_{ij} =& \underbrace{\Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{m}}_j - \Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{s}}_i - \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{t}}}_{\boldsymbol{u}_{ij}},\end{aligned}$$

where $\Omega_1(\boldsymbol{q})^T\Omega_2(\boldsymbol{q}) = \widehat{\boldsymbol{R}}$, and $\Omega_1(\boldsymbol{q})\Omega_1(\boldsymbol{q})^T = \boldsymbol{I}_{4\times 4}$ (more properties can be obtained from Chapter 2).

Let $\boldsymbol{m}_j^o, \boldsymbol{s}_i^o, \boldsymbol{t}^o$ be the noise-free model point, sensor point and translation estimate respectively. We have $\boldsymbol{m} = \boldsymbol{m}^o + \delta\boldsymbol{m}$, $\boldsymbol{s}_i = \boldsymbol{s}_i^o + \delta\boldsymbol{s}_i$, $\boldsymbol{t} = \boldsymbol{t}^o + \delta\boldsymbol{t}$, where $\delta\boldsymbol{m}, \delta\boldsymbol{s}_i, \delta\boldsymbol{t}$ are the respective noise in model point, sensor point and translation vector. All the noises are assumed to be sampled from zero mean Gaussian distributions with covariances $\boldsymbol{\Sigma}_{\boldsymbol{m}_j}, \boldsymbol{\Sigma}_{\boldsymbol{s}_i}, \boldsymbol{C}$ respectively. When there is no noise, the transformed noise-free sensor point must be located at the same place as the corresponding noise-free model point,

i.e., $\widetilde{\boldsymbol{r}}_{ij} = 0$,

$$\boldsymbol{u}_{ij} - \underbrace{\left(\Omega_1(\boldsymbol{q})\widetilde{\delta\boldsymbol{m}_j} - \Omega_2(\boldsymbol{q})\widetilde{\delta\boldsymbol{s}_i} - \Omega_1(\boldsymbol{q})\widetilde{\delta\boldsymbol{t}}\right)}_{\delta\boldsymbol{u}_{ij}} = 0, \tag{9.21}$$

where $\delta\boldsymbol{u}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{S}_{ij})$, and the expression for $\boldsymbol{S}_{ij}$ can be obtained in closed form using the result of Proposition 1,

$$\begin{aligned}\boldsymbol{S}_{ij} =& \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{\Sigma}_{\boldsymbol{m}_j}}\Omega_1(\boldsymbol{q})^T + \Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{\Sigma}_{\boldsymbol{s}_i}}\Omega_2(\boldsymbol{q})^T + \\ & \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{C}}\Omega_1(\boldsymbol{q})^T + \boldsymbol{A}_1(\widetilde{\boldsymbol{\Sigma}_{\boldsymbol{m}_j}} \circledast \boldsymbol{D})\boldsymbol{A}_1^T + \\ & \boldsymbol{A}_2(\widetilde{\boldsymbol{\Sigma}_{\boldsymbol{s}_i}} \circledast \boldsymbol{D})\boldsymbol{A}_2^T + \boldsymbol{A}_3(\widetilde{\boldsymbol{C}} \circledast \boldsymbol{D})\boldsymbol{A}_3^T.\end{aligned} \tag{9.22}$$

Using Eq. 9.20, we derive the form of $P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x})$,

$$P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x}) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{u}_{ij}^T(\boldsymbol{S}_{ij})^{-1}\boldsymbol{u}_{ij}\right)}{\sqrt{(2\pi)^4|\boldsymbol{S}_{ij}|}}, \tag{9.23}$$

where $\boldsymbol{u}_{ij}$ and $\boldsymbol{S}_{ij}$ are obtained from Eq. 9.21 and Eq. 9.22 respectively. Substituting Eq. 9.23 in Eq. 9.11,

$$w_{ij} = \frac{\exp(-\frac{1}{2}\boldsymbol{u}_{ij}^T(\boldsymbol{S}_{ij})^{-1}\boldsymbol{u}_{ij})}{\sqrt{|\boldsymbol{S}_{ij}|}\sum_{k=1}^{n_{\mathcal{M}}}\frac{\exp(-\frac{1}{2}\boldsymbol{u}_{ik}^T(\boldsymbol{S}_{ik})^{-1}\boldsymbol{u}_{ik})}{\sqrt{|\boldsymbol{S}_{ik}|}}}. \tag{9.24}$$

Substituting Eq. 9.20, 9.23, 9.24 in Eq. 9.12 and discarding terms independent of $\boldsymbol{x}$ gives

$$\begin{aligned}O(\boldsymbol{x}, W) =& -\boldsymbol{q}^T\boldsymbol{D}\boldsymbol{q} + (\boldsymbol{t}-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{t}-\boldsymbol{\mu}) \\ & + \sum_{i=1}^{n_{\mathcal{S}}}\sum_{j=1}^{n_{\mathcal{M}}} w_{ij}\left(\boldsymbol{u}_{ij}^T(\boldsymbol{S}_{ij})^{-1}\boldsymbol{u}_{ij} - \log\left(|\boldsymbol{S}_{ij}|\right)\right).\end{aligned} \tag{9.25}$$

$O(\boldsymbol{x}, W)$ is minimized in a nested fashion, by first optimizing over $\boldsymbol{q}$ followed by $\boldsymbol{t}$ and repeating the process until convergence. It is difficult to optimize $O(\boldsymbol{x}, W)$ with the log term in it, as it makes the equation highly non-linear. However, if one were to

Input:
$\mathcal{S} = \{\boldsymbol{s}_i \in \mathbb{R}^3\}, i = 1, 2, ..., n_{\mathcal{S}}$
$[\mathcal{M} = \{\boldsymbol{m}_j \in \mathbb{R}^3\}, j = 1, 2, ..., n_{\mathcal{M}}]$
(Surface model $\mathcal{M}$)
Initial transformation: $\boldsymbol{q}_0 \in \mathbb{R}^4, \boldsymbol{t}_0 \in \mathbb{R}^3$,
Prior pose uncertainty: $\boldsymbol{D}_0 \in \mathbb{R}^{4\times4}$, $\boldsymbol{C}_0 \in \mathbb{R}^{3\times3}$
Output:
$\boldsymbol{q} \in \mathbb{R}^4, \boldsymbol{t} \in \mathbb{R}^3$ that aligns $\mathcal{S}$ and $\mathcal{M}$
$\boldsymbol{D} \in \mathbb{R}^{4\times4}, \boldsymbol{C} \in \mathbb{R}^{3\times3}$
Initialize: $k = 1$
while *not converged* **do**
  Correspondence:
  $[w_{ij} = E((W_{\boldsymbol{x}})_{ij})$ from Eq. 9.11]
  (Update $\boldsymbol{m}_i$, $i = 1, \ldots, n_{\mathcal{S}}$ from Eq. 9.33)
  Minimization:
  Initialize: $l = 1$
  while *not converged* **do**
    Initialize: $m = 1$
    while *not converged* **do**
      Update $\boldsymbol{q}$ [from Eq. 9.26] (from Eq. 9.35)
      Update $\boldsymbol{S}_{ij}$ [from Eq. 9.22] (from Eq. 9.22 after substituting $\boldsymbol{\Sigma}_{\boldsymbol{m}_j} = \boldsymbol{0}$)
      $m = m + 1$
    end
    Update $\boldsymbol{t}$ [using Eq. 9.27] (using Eq. 9.36)
    $l = l + 1$
  end
  $\boldsymbol{q}_k = \boldsymbol{q}$
  $\boldsymbol{t}_k = \boldsymbol{t}$
  Update $\boldsymbol{D}_k$ using [Eq. 9.30] (Eq. 9.37)
  Update $\boldsymbol{C}_k$ using [Eq. 9.31] (Eq. 9.38)
  $k = k + 1$
end
$\boldsymbol{q} = \boldsymbol{q}_k$, $\boldsymbol{t} = \boldsymbol{t}_k$, $\boldsymbol{D} = \boldsymbol{D}_k$, $\boldsymbol{C} = \boldsymbol{C}_k$

Algorithm 5: Point cloud to [point cloud] (surface model) registration

use a branch and bound technique such as Go-ICP [18], the log term could be easily considered. But in this chapter, we neglect the log term.

We first optimize $O(\boldsymbol{x}, W)$ in Eq. 9.25 over $\boldsymbol{q}$ while also imposing the constraint $\boldsymbol{q}^T\boldsymbol{q} = 1$ with a Lagrange multiplier

$$\begin{aligned}
&\frac{\partial O(\boldsymbol{x}, W) - \lambda(\boldsymbol{q}^T\boldsymbol{q} - 1)}{\partial \boldsymbol{q}} = 0, \\
\Rightarrow &\left(\underbrace{-\boldsymbol{D} + \sum_{i=1}^{n_{\mathcal{S}}} \sum_{j=1}^{n_{\mathcal{M}}} w_{ij} \boldsymbol{H}_{ij}^T (\boldsymbol{S}_{ij})^{-1} \boldsymbol{H}_{ij}}_{\boldsymbol{B}_0} - \lambda \boldsymbol{I}_{4\times 4} \right) \boldsymbol{q} = 0, \\
\Rightarrow &\left(\boldsymbol{B}_0 - \lambda \boldsymbol{I}_{4\times 4}\right) \boldsymbol{q} = 0,
\end{aligned} \tag{9.26}$$

where $\boldsymbol{H}_{ij} = \Omega_2(\widetilde{\boldsymbol{m}}_j) - \Omega_1(\widetilde{\boldsymbol{s}}_i) - \Omega_2(\widetilde{\boldsymbol{t}})$. $\boldsymbol{q}$ can be obtained from Eq. 9.26 by solving an eigen value problem, with λ being the smallest eigen value and $\boldsymbol{q}$ being the eigen vector corresponding to the smallest eigen value of $\boldsymbol{B}_0$. Note that from Eq. 9.22, $\boldsymbol{S}_{ij}$ is also dependent on $\boldsymbol{q}$ and we use the current best estimate of $\boldsymbol{q}$ to estimate $\boldsymbol{S}_{ij}$. Upon obtaining $\boldsymbol{q}$, we update $\boldsymbol{S}_{ij}$ and once again update $\boldsymbol{q}$ from Eq. 9.26. This process is repeated until convergence or a maximum number of iterations are reached.

After obtaining the best estimate of $\boldsymbol{q}$ given $\boldsymbol{t}$, we keep the $\boldsymbol{q}$ constant and optimize for $\boldsymbol{t}$,

$$\begin{aligned}
&\frac{\partial O(\boldsymbol{x}, W)}{\partial \boldsymbol{t}} = 0, \\
\Rightarrow &\boldsymbol{C}^{-1}(\boldsymbol{t} - \boldsymbol{\mu}) - \sum_{i=1}^{n_{\mathcal{S}}} \sum_{j=1}^{n_{\mathcal{M}}} w_{ij} \overline{\Omega_1(\boldsymbol{q})}^T (\boldsymbol{S}_{ij})^{-1} (\boldsymbol{u}_{ij}) = 0, \\
\Rightarrow &\boldsymbol{t} = \left(\boldsymbol{C}^{-1} + \boldsymbol{B}_1\right)^{-1} \left(\boldsymbol{C}^{-1}\boldsymbol{\mu} + \boldsymbol{B}_2\right), \quad \text{where}
\end{aligned} \tag{9.27}$$

$$\boldsymbol{B}_1 = \sum_{i=1}^{n_{\mathcal{S}}} \sum_{j=1}^{n_{\mathcal{M}}} w_{ij} \overline{\Omega_1(\boldsymbol{q})}^T (\boldsymbol{S}_{ij})^{-1} \overline{\Omega_1(\boldsymbol{q})}, \tag{9.28}$$

$$\boldsymbol{B}_2 = \sum_{i=1}^{n_{\mathcal{S}}} \sum_{j=1}^{n_{\mathcal{M}}} w_{ij} \overline{\Omega_1(\boldsymbol{q})}^T \boldsymbol{S}_{ij}^{-1} (\overline{\Omega_1(\boldsymbol{q})} \boldsymbol{m}_j - \overline{\Omega_2(\boldsymbol{q})} \boldsymbol{s}_i). \tag{9.29}$$

After obtaining $\boldsymbol{q}$ and $\boldsymbol{t}$ iteratively from Eq. 9.26 and Eq. 9.27, we update the

uncertainties in $\boldsymbol{q}$ and $\boldsymbol{t}$ by taking the double derivative of $O(\boldsymbol{x}, W)$ with respect to $\boldsymbol{q}$ and $\boldsymbol{t}$,

$$\boldsymbol{D}_{k+1} = \boldsymbol{D}_k - \sum_{i,j} w_{ij} \boldsymbol{H}_{ij}^T (\boldsymbol{S}_{ij})^{-1} \boldsymbol{H}_{ij}, \tag{9.30}$$

$$\boldsymbol{C}_{k+1} = \left(\boldsymbol{C}_k^{-1} + \sum_{i,j} w_{ij} \overline{\Omega_1(\boldsymbol{q})}^T (\boldsymbol{S}_{ij})^{-1} \overline{\Omega_1(\boldsymbol{q})} \right)^{-1}, \tag{9.31}$$

where k is the index of the outermost iteration. A pseudo code describing the optimization is provided in Alg. 5.

9.3.2 Case 2:Registering point cloud to a surface model

Substituting Eq. 9.20 and Eq. 9.23 in Eq. 9.16 and discarding terms independent of $\boldsymbol{x}, \overline{\boldsymbol{m}}$ gives

$$\begin{aligned} O(\boldsymbol{x}, \overline{\boldsymbol{m}}) = & - \boldsymbol{q}^T \boldsymbol{D} \boldsymbol{q} + (\boldsymbol{t} - \boldsymbol{\mu})^T \boldsymbol{C}^{-1} (\boldsymbol{t} - \boldsymbol{\mu}) \\ & + \sum_{i=1}^{n_{\mathcal{S}}} \left(\boldsymbol{u}_{ii}^T (\boldsymbol{S}_i)^{-1} \boldsymbol{u}_{ii} - \log \left(|\boldsymbol{S}_i| \right) \right), \end{aligned} \tag{9.32}$$

where $\boldsymbol{S}_i$ is obtained by substituting $\boldsymbol{\Sigma}_{\boldsymbol{m}_j} = \boldsymbol{0}_{3\times 3}$ in Eq. 9.22.

We find the optimal value for $\boldsymbol{x}$ and $\overline{\boldsymbol{m}}$ in an iterative manner. The corresponding model points $\overline{\boldsymbol{m}}_{\boldsymbol{x}_k}$, given the current best estimate $\boldsymbol{x}_k$ is obtained as shown below

$$\boldsymbol{m}_i = \underset{\boldsymbol{m} \in \mathcal{M}}{\operatorname{argmin}} \, \boldsymbol{u}_i^T (\boldsymbol{S}_i)^{-1} \boldsymbol{u}_i, \tag{9.33}$$

where $\boldsymbol{u}_i = \Omega_1(\boldsymbol{q}) \widetilde{\boldsymbol{m}} - \Omega_2(\boldsymbol{q}) \widetilde{\boldsymbol{s}}_i - \Omega_1(\boldsymbol{q}) \widetilde{\boldsymbol{t}}$. Unlike in the case of registering two point clouds, the covariance corresponding to model points $\boldsymbol{m}$ on the surface are set to zero. This constrains the model points to lie on the surface. Eq. 9.33 reduces to finding closest points using a Mahalanobis distance metric as shown below. Let us consider

the following definition for d

$$\begin{aligned} d &= \boldsymbol{u}_i^T (\boldsymbol{S}_i)^{-1} \boldsymbol{u}_i, \\ &= \left(\Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{m}} - \Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{s}}_i - \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{t}}\right)^T (\boldsymbol{S}_i)^{-1} \left(\Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{m}} - \Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{s}}_i - \Omega_1(\boldsymbol{q})\widetilde{\boldsymbol{t}}\right), \\ &= \widetilde{\boldsymbol{d}}_i^T \left(\Omega_1(\boldsymbol{q})^T \boldsymbol{S}_i^{-1} \Omega_1(\boldsymbol{q})\right) \widetilde{\boldsymbol{d}}_i, \end{aligned} \tag{9.34}$$

where $\widetilde{\boldsymbol{d}}_i = \widetilde{\boldsymbol{m}} - \widetilde{\boldsymbol{s}_i^{\mathcal{M}}} - \widetilde{\boldsymbol{t}}$, $\widetilde{\boldsymbol{s}}_i = \Omega_1^T(\boldsymbol{q})\Omega_2(\boldsymbol{q})\widetilde{\boldsymbol{s}}_i$. $\boldsymbol{s}_i^{\mathcal{M}}$ is the transformed location of $\boldsymbol{s}_i$. From Eq. 9.34, d is the Mahalanobis distance between $\widetilde{\boldsymbol{m}}$ and $\widetilde{\boldsymbol{s}_i^{\mathcal{M}}}$, with covariance matrix $\Omega_1(\boldsymbol{q})^T \boldsymbol{S}_i \Omega_1(\boldsymbol{q})$.

As in the case of registration of point clouds, the optimal registration parameters are obtained in an iterative fashion for the case of registration to surface model.

The optimal value of $\boldsymbol{q}$ assuming $\boldsymbol{t}$ to be fixed is obtained by solving for $\boldsymbol{q}$ in

$$(\boldsymbol{F}_0 - \lambda \boldsymbol{I}_{4\times 4})\boldsymbol{q} = \boldsymbol{0}, \tag{9.35}$$

where $\boldsymbol{F}_0$ is equivalent to $\boldsymbol{B}_0$ derived in Eq. 9.26 with the substitution of $wij = \delta_{ij}$ (δ_{ij} is the Kronecker delta). $\boldsymbol{q}$ can be obtained by finding the eigen vector corresponding to the smallest eigen value of $\boldsymbol{F}_0$.

The optimal value of $\boldsymbol{t}$ assuming $\boldsymbol{q}$ to be fixed is

$$\Rightarrow \boldsymbol{t} = \left(\boldsymbol{C}^{-1} + \boldsymbol{F}_1\right)^{-1} \left(\boldsymbol{C}^{-1}\boldsymbol{\mu} + \boldsymbol{F}_2\right), \tag{9.36}$$

where $\boldsymbol{F}_1$, $\boldsymbol{F}_2$ are equivalent to $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ obtained from Eq. 9.28 and Eq. 9.29 respectively with the substitution $wij = \delta_{ij}$.

After obtaining $\boldsymbol{q}$ and $\boldsymbol{t}$ iteratively from Eq. 9.35 and Eq. 9.36, we update the uncertainties in $\boldsymbol{q}$ and $\boldsymbol{t}$,

$$\boldsymbol{D}_{k+1} = \boldsymbol{D}_k - \sum_{i=1}^{n_{\mathcal{S}}} \boldsymbol{H}_{ii}^T (\boldsymbol{S}_i)^{-1} \boldsymbol{H}_{ii}, \tag{9.37}$$

$$\boldsymbol{C}_{k+1} = \left(\boldsymbol{C}_k^{-1} + \sum_{i=1}^{n_{\mathcal{S}}} \overline{\Omega_1(\boldsymbol{q})}^T (\boldsymbol{S}_i)^{-1} \overline{\Omega_1(\boldsymbol{q})}\right)^{-1}, \tag{9.38}$$

where k is the index of the outermost iteration. A pseudo code describing the optimization is provided in Alg. 5. In Alg. 5, the entities within $(\cdot)$ correspond to registering to a point cloud, while the entities within $[\cdot]$ correspond to registering to a surface model.

9.4 Generalization to Other Prior Approaches

9.4.1 Iterative Closest Point

ICP was originally developed for registering a data shape (which is decomposed into a point set if not already in that form) to a model shape [11]. We consider our approach for Case 2 and show that the ICP is a specific case of our implementation.

Let us make the following assumptions:

- Equal and isotropic noise, $\boldsymbol{\Sigma}_{\boldsymbol{s}_i} = \lambda_1 \boldsymbol{I}_{3\times 3}$.
- No knowledge of prior distribution, hence we will resort to MLE.

Substituting the above assumptions in Eq. 9.22, gives $\boldsymbol{S}_i^{-1} = \frac{1}{\lambda_1}\boldsymbol{I}_{4\times 4}$. Substituting $\mathcal{S}_i$ in Eq. 9.33 and simplifying gives,

$$\boldsymbol{m}_i = \underset{\boldsymbol{m}\in\mathcal{M}}{\operatorname{argmin}} (\boldsymbol{m} - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t})^T (\boldsymbol{m} - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t}),$$

which is the closest point criteria used to find correspondences in ICP.

Substituting $\boldsymbol{S}_i$ in Eq. 9.36 and simplifying gives,

$$\boldsymbol{t} = \boldsymbol{m}_c - \boldsymbol{R}\boldsymbol{s}_c, \tag{9.39}$$

which is the same expression used to find $\boldsymbol{t}$ in the case of ICP. Substituting $\boldsymbol{t}$ from Eq. 9.39 into Eq. 9.35

$$\boldsymbol{B}_0 = \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} (\Omega_2(\widetilde{\boldsymbol{m}}_i - \widetilde{\boldsymbol{m}}_c) - \Omega_1(\widetilde{\boldsymbol{s}}_i - \widetilde{\boldsymbol{s}}_c)). \tag{9.40}$$

$\boldsymbol{q}$ is obtained by finding the eigen vector corresponding to the smallest eigen value of $\boldsymbol{B}_0$, which is the same as using the approach of Faugeras and Hebert [60] in ICP.

Thus, when performing MLE for registering a point cloud with equal isotropic noise to a surface model, our approach becomes equivalent to ICP and it is the optimal estimator.

9.4.2 EM-ICP

EM-ICP is similar to ICP, with the main difference being that it was developed for registering point cloud to point cloud [241, 85]. Myronenko *et. al.* [116] developed an approach called coherent point drift (CPD), which is very similar to the EM-ICP. The important difference between CPD and EM-ICP is the method chosen to handle the outliers. EM-ICP uses *structural editing*, a technique to remove outliers, instead of using a probabilistic weight to account for outliers as in the case of CPS [116]. Ignoring the approach to handle outliers, bth these methods are identical and make the same assumptions and hence we will refer to both of them as EM-ICP in this chapter. We make the following assumptions

- $\boldsymbol{\Sigma}_{\boldsymbol{s}_i} = \lambda_1 \boldsymbol{I}_{3\times 3}$ and $\boldsymbol{\Sigma}_{\boldsymbol{m}_i} = \boldsymbol{0}_{3\times 3}$.
- No knowledge of prior distribution, hence we will resort to MLE.

Upon considering our assumptions, Eq. 9.23 becomes

$$P(\boldsymbol{s}_i|\boldsymbol{m}_j, \boldsymbol{x}) = \frac{-\exp(\frac{1}{2}\boldsymbol{d}_{ij}^T \boldsymbol{d}_{ij})}{\lambda_1 \sqrt{(2\pi\lambda_1)^4}}, \tag{9.41}$$

where $\boldsymbol{d}_{ij} = \boldsymbol{m}_j - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t}$. It can be verified that upon substituting Eq. 9.41 in Eq. 9.11, Eq. 9.12, the equations for expectation and minimization are identical to the ones in EM-ICP.

Thus, when performing MLE for registering a point cloud with equal isotropic noise to another point cloud, our approach becomes equivalent to EM-ICP and it is the optimal estimator.

9.4.3 Generalized Total-Least-Squares ICP

Generalized Total-Least-Squares ICP (GTLS-ICP) and generalized ICP (G-ICP) are extensions of ICP that takes into account anisotropic noise in model and sensed points [21, 22]. As with ICP, both GTLS-ICP and G-ICP are MLE methods.

GTLS-ICP and G-ICP, make an approximation when it comes to the expectation step. Instead of using soft matches using Eq. 9.11, the authors use a closest point rule, similar to ICP.

Assuming MLE, Eq. 9.22 can be rearranged to obtain

$$\Omega_1(\boldsymbol{q})^T \boldsymbol{S}_{ij} \Omega_1(\boldsymbol{q}) = \widetilde{\boldsymbol{\Sigma}_{\boldsymbol{m}_j}} + \widehat{\boldsymbol{R}} \widetilde{\boldsymbol{\Sigma}_{\boldsymbol{s}_i}} \widehat{\boldsymbol{R}}^T. \tag{9.42}$$

Upon substituting Eq. 9.42 in Eq. 9.12 and ignoring the log term gives

$$O(\boldsymbol{x}, W) = \sum_{i=1}^{n_{\mathcal{S}}} \boldsymbol{d}_i^T \left(\boldsymbol{\Sigma}_{\boldsymbol{m}_{c(i)}} + \boldsymbol{R} \boldsymbol{\Sigma}_{\boldsymbol{s}_i} \boldsymbol{R}^T \right)^{-1} \boldsymbol{d}_i, \tag{9.43}$$

where $\boldsymbol{d}_i = \boldsymbol{m}_{c(i)} - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t}$ and $w_{ij} = \delta_{c(i)\ j}$ (δ is the Kronecker delta). The objective function in Eq. 9.43 is identical to the ones used in GTLS-ICP and G-ICP [21, 22].

Another variation of the GTLS-ICP is the A-ICP [244]. The A-ICP uses a Mahalanobis distance metric instead of closest point rule. This approximation is shown to provide an improvement over GTLS-ICP and G-ICP [244].

9.4.4 Iterative Most Likely Point Registration

The iterative most likely point (IMLP) is also a MLE and it was developed as an improvement over A-ICP and other GTLS methods [23]. When registering point clouds, IMLP is very similar to A-ICP, with the important distinction being the definition of w_{ij}. Unlike A-ICP, IMLP computes w_{ij} using Eq. 9.11 without any approximations. However, after computing w_{ij}, they set the w_{ij} with highest value to 1 and the rest to 0.

When registering to a surface model, the correspondence is obtained from Eq. 9.33,

$$\boldsymbol{m}_i = \underset{\boldsymbol{m}\in\mathcal{M}}{\operatorname{argmin}}\, \boldsymbol{d}_i^T(\boldsymbol{\Sigma}_{\boldsymbol{m}_i} + \boldsymbol{R}\boldsymbol{\Sigma}_{\boldsymbol{s}_i}\boldsymbol{R}^T)^{-1}\boldsymbol{d}_i,$$

where $\boldsymbol{d}_i = \boldsymbol{m}_i - \boldsymbol{R}\boldsymbol{s}_i - \boldsymbol{t}$. Thus the correspondence criteria is identical to the correspondence criteria of IMLP for registering point cloud to surface model.

The objective function for minimization is obtained from Eq. 9.32,

$$\begin{aligned} O(\boldsymbol{x}, \overline{\boldsymbol{m}}) &= \sum_{i=1}^{n_{\mathcal{S}}} \left(\boldsymbol{u}_{ii}^T(\boldsymbol{S}_i)^{-1}\boldsymbol{u}_{ii} - \log(|\boldsymbol{S}_i|)\right), \\ &= \sum_{i=1}^{n_{\mathcal{S}}} [\boldsymbol{d}_i^T(\boldsymbol{\Sigma}_{\boldsymbol{m}_i} + \boldsymbol{R}\boldsymbol{\Sigma}_{\boldsymbol{s}_i}\boldsymbol{R}^T)^{-1}\boldsymbol{d}_i - \\ &\qquad \log(|\boldsymbol{\Sigma}_{\boldsymbol{m}_i} + \boldsymbol{R}\boldsymbol{\Sigma}_{\boldsymbol{s}_i}\boldsymbol{R}^T|)], \end{aligned} \tag{9.44}$$

The objective function in Eq. 9.44 is identical to the one used in IMLP.

Thus when registering to a surface model and assuming MLE, our approach is identical to IMLP. However, when registering to a point cloud, our approach uses expectation of matches, while IMLP uses most likely matches. As shown by other soft matching methods, our approach is expected to have a wider basin of convergence when compared to IMLP.

9.4.5 Extended, Iterative Extended and Unscented Kalman Filter for Registration

Unlike the methods discussed thus far, the EKF, IEKF and UKF-based registration approach considers prior distribution in the registration parameters. All the filtering-based approaches use a MAP estimate as shown in Eq. 9.2. However, there are two key differences

- Different filtering approaches use different parameterizations for pose. For example, Moghari *et. al.* [5] use Euler angles and Cartesian coordinates, while Hauberg *et. al.* [49] use Lie algebra elements to parameterize the pose.

- The optimization method used is different for different filters. EKF is a special case of Gauss Newton optimization without iterations [246], IEKF is a Gauss Newton update [247] and UKF is a population-based stochastic optimization technique [181].

All the filtering-based methods use a closest point matching rule. The objective function they use is identical to Eq. 9.18. However, the actual form of the objective function differs due to the choice of parameterization used by each method.

9.4.6 Bingham Filter-based Registration

Among the filtering-based methods, the one that comes closest our approach is the Bingham filter-based registration described in Chapter 6. This approach makes all the assumptions that we make in this chapter. An important distinctions is that the bingham filter uses a closest point matching when registering to a point cloud as well as to a surface model.

9.4.7 Suggestions For Improvement

1. ICP and EM-ICP assume equal isotropic noise in sensed points and no noise in model points. We can easily relax these assumptions with our approach.

2. GTLS-ICP, G-ICP, A-ICP and IMLP use an approximate function for finding the expectation of point matching when registering point clouds. Our approach can provide more accurate point matching expectation and hence improve the registration.

3. All the filtering-based approaches use an approximate closest point matching. Our approach can use a more accurate point matching.

9.5 Results

In this section we present experimental results for our approach and compare the same with other prior approaches. We first present results for the case where the correspondence is known, and then present results for the case with unknown correspondence. For both the cases we present results for MLE and MAP. For the case of unknown correspondence we present results for registering point cloud as well as surface model. We terminate all the algorithms when the number of iterations exceed 100 or when the change in translation is less than 10^{-4}mm and change in rotation is less than 10^{-4}deg. We start all algorithms with an initial condition of $\boldsymbol{q} = (1, 0, 0, 0)^T$ and $\boldsymbol{t} = (0, 0, 0)^T$.

Table 9.3: Experimental Results

	x (mm)	y (mm)	z (mm)	θ_x (deg)	θ_y (deg)	θ_z (deg)	RMS (mm)
Results for Experiment 1: Isotropic Noise							
Actual	13.13	-48.53	-67.59	-19.29	11.33	-3.59	–
ICP, EM-ICP	13.15	-48.59	-67.58	-19.25	11.31	-3.60	0.0038
IMLP	13.15	-48.59	-67.58	-19.25	11.31	-3.60	0.0038
Our approach	13.15	-48.59	-67.58	-19.25	11.31	-3.60	0.0038
Results for Experiment 1: Anisotropic Noise							
ICP, EM-ICP	13.17	-48.51	-67.59	-19.25	11.29	-3.59	0.003
IMLP	13.16	-48.52	-67.59	-19.25	11.33	-3.58	0.002
Our approach	13.18	-48.53	-67.58	-19.25	11.32	-3.58	0.002
Results For Experiment 3							
Actual	24.90	-18.82	19.14	6.78	16.18	15.73	–
EM-ICP	24.04	-18.34	18.86	6.17	15.69	15.61	0.1531
GTLS-ICP	24.16	-18.46	18.97	5.98	15.52	15.58	0.1204
A-ICP	23.83	-17.99	19.06	6.37	15.76	15.91	0.1223
IMLP	24.03	-18.10	19.10	6.43	15.73	15.93	0.1160
Our approach	24.73	-18.65	18.91	7.25	15.84	15.94	0.0575
Results For Experiment 4							
Actual	-0.59	-24.84	9.56	13.36	2.08	15.07	–
ICP	0.67	-24.98	9.10	12.83	1.52	14.79	0.1167
IMLP	-0.11	-24.34	8.84	12.76	1.72	15.26	0.0732
Our approach	-0.06	-24.24	8.86	12.71	1.54	15.44	0.0623

9.5.1 Experiment 1: Known Correspondence MLE

In this experiment we first study the case of isotropic noise and then study anisotropic noise in the points. When the correspondence is known, both ICP and EM-ICP are identical. GTLS-ICP, G-ICP, A-ICP and IMLP have identical objective functions, but use different optimizers. Billings *et. al.* [23] show that IMLP outperforms GTLS-ICP, G-ICP and A-ICP when the correspondence is known and hence we skip those comparisons to save space.

The studies in this experiment were conducted by generating two noisy point sets with known correspondence and known ground-truth alignment. For groundtruth, we randomly generated 100 points uniformly distributed in the interval [-100, 100]mm along each dimension. From these points, two different noisy point sets were generated by adding zero-mean Gaussian noise, while using a different covariance for each point set. The eigen values of the noise covariance we set to $[5, 5, 5]\text{mm}^2$ for the case of isotropic noise and $[5, 5, 20]\text{mm}^2$ for the case of anisotropic noise. Registration accuracy is evaluated by computing the mean distance between the noise-free point correspondences following each registration. Table 9.3 shows the results for a representative case. When the noise is isotropic, all methods have identical performance. When the noise is anisotropic, IMLP and our approach was identical (as was shown theoretically in Sec. 9.4) and performed better than ICP.

9.5.2 Experiment 2: Known Correspondence MAP

The setup for this experiment is identical to experiment 1, with the only difference being that instead of using all the measurements at once, we use a mini-batches of last 10 measurements in each update. After each update, we take the current best estimate of registration as the prior and estimate the next best estimate. For MAP with known correspondence and anisotropic noise, the Bingham filter is identical to our approach. In a similar experimental setup, in Chapter 6, we show that the Bingham filter outperforms the EKF and UKF-based registration approaches and hence we do not repeat those comparisons in this work. While the ICP and IMLP

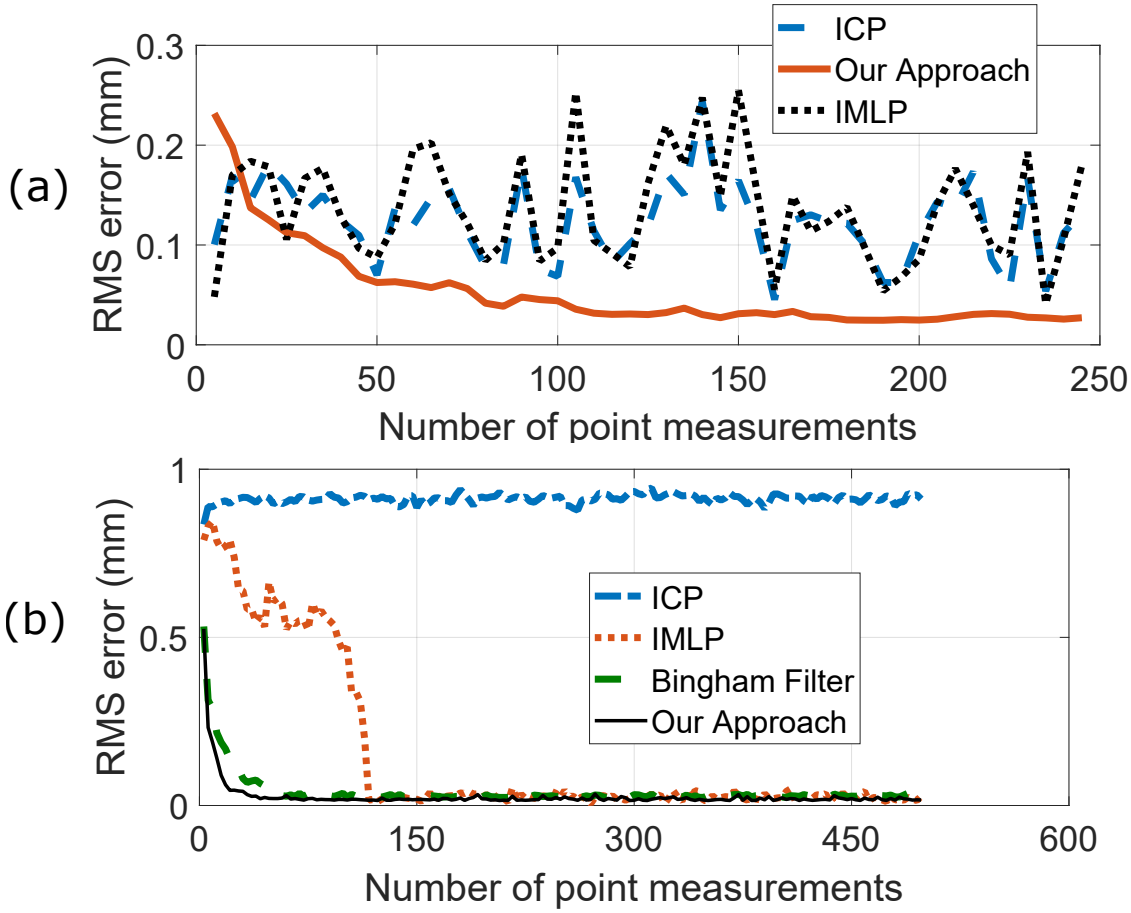


Figure 9-2: (a) Results for Experiment 2. (b) Results for Experiment 5.

are not MAP approaches, we use them repeatedly to update the registration after each small batch of measurements using the previous estimate as the starting point. Fig. 9-2(a) shows the RMS error vs the number of point measurements used. Our approach is effective at filtering out the noise and consistently having a lower RMS error compared to the other approaches.

9.5.3 Experiment 3: Unknown Correspondence MLE and Registration to a Point Cloud

The experimental setup for Experiment 3 is identical to the anisotropic noise case of Experiment 1. We calculate the RMS error by calculating the average distance between the registered and known ground-truth positions of the points. Table 9.3 shows the results for this experiment. EM-ICP performs the worst of all the algorithms because it does not handle anisotropic noise. As expected, IMLP outperforms EM-ICP, GTLS-ICP and A-ICP. Our approach offers further improvement over IMLP, which is enabled by the soft matching criteria as opposed to a hard most likely match of IMLP (See theoretical study in Sec. 9.4).

9.5.4 Experiment 4: Unknown Correspondence MLE and Registration to a Surface Model

For this experiment, we consider a Stanford bunny [7] for the surface model (see Fig. 9-3). The experiments are conducted by randomly generating a set of 100 noisy points from the mesh surface and applying a random misalignment to the points. The anisotropic noise applied is the same as in Experiment 1. We evaluate ICP, IMLP and our approach over this data as shown in Table 9.3. We measure the registration error by randomly sampling a set of 100 noise-free points from the mesh and finding the mean distance between the registered and known ground-truth positions of the points. Once again our approach achieves lower RMS error than ICP and IMLP.

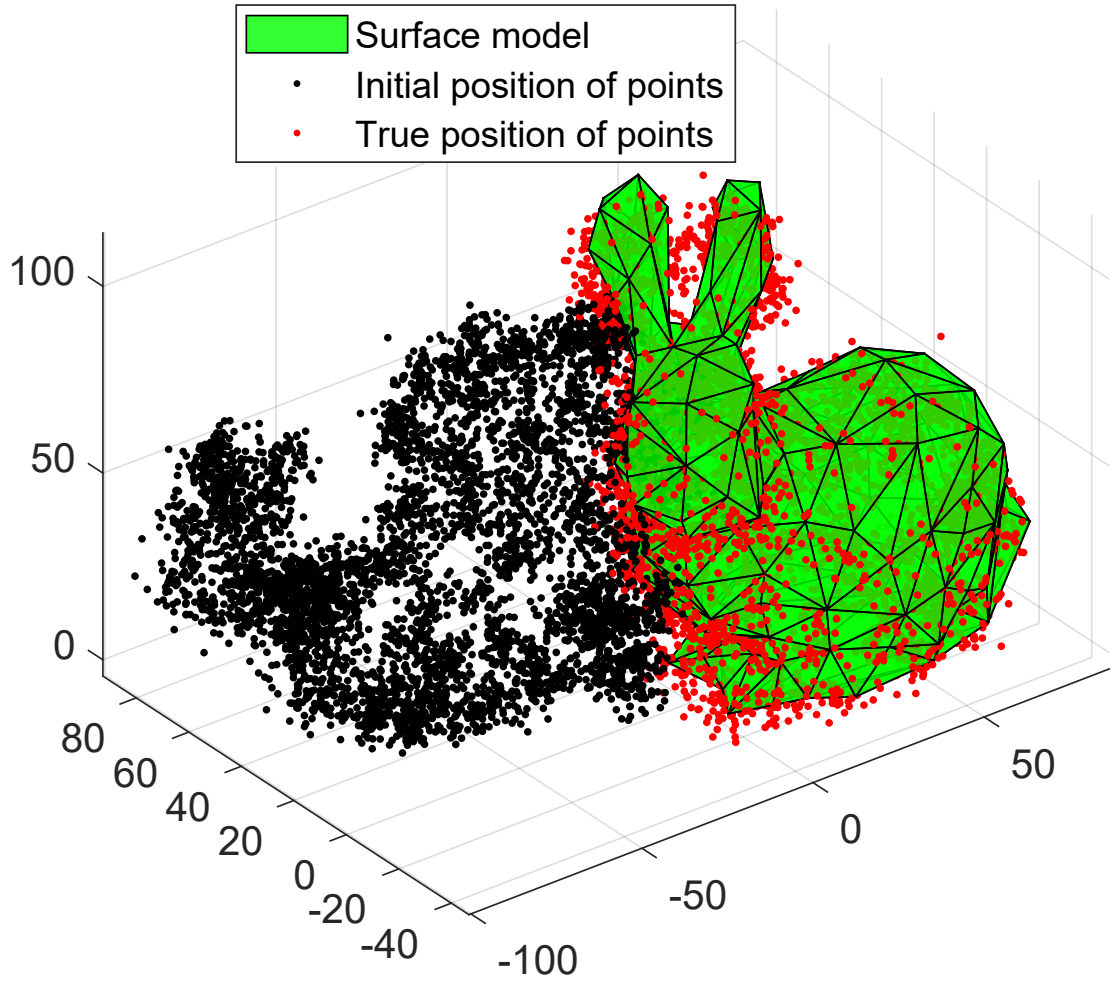


Figure 9-3: Triangulated mesh model of Stanford bunny is shown in green. Black points represent initial location and red points represent the registered location of points.

9.5.5 Experiment 5: Unknown Correspondence MAP and Registration to a Surface Model

We use the same setup as Experiment 4, but update registration after each mini-batch of 30 measurements. After each update, we take the current best estimate of registration as the prior and estimate the next best estimate. For both the Bingham filter as well as our approach we assume large initial uncertainty in distribution of pose– $\boldsymbol{D}_0 = \epsilon \boldsymbol{I}_{4\times4}$ and $\boldsymbol{C}_0^{-1} = \epsilon \boldsymbol{I}_{3\times3}$, where $\epsilon \to 0$. Fig. 9-2(b) shows the RMS error vs the number of point measurements used. Our approach converges the fastest among all the approaches and also converges to the lowest value. The Bingham filter being a MAP estimate performs better than IMLP. The RMS error after processing 500 points is 0.91mm for ICP, 0.027mm for IMLP, 0.026mm for Bingham filter and 0.016mm for our approach.

9.6 Conclusion

Despite decades of research in the field of point registration, it is still a challenge to decide which algorithm to pick for a given application. The work presented in this chapter unifies a variety of existing registration methods into a framework whose parameters can be intuitively chosen to obtain the optimal approach for a given application. Furthermore, it is observed that different research groups approach the problem of point registration differently. This framework unites a lot of them in a single optimization framework with different prior approaches being special cases of the overlaying framework.

A noteworthy outcome of this framework is the ability to predict deficiencies in existing approaches. To this effect we provide means to improve a number of methods and validate the efficacy of the improvement in experiments.

While this framework provides a probabilistic view of the point registration problem, it does not guarantee globally optimal solutions; much like the existing approaches that it generalizes. However, the objective functions derived, can be min-

imized without the approximations made, using global optimizers. This is a task for future work. This task would also allow us to compare our results with more simplistic global registration methods in literature. Future work also involves extending the framework to accommodate surface normals, curvature, and other geometric descriptors.

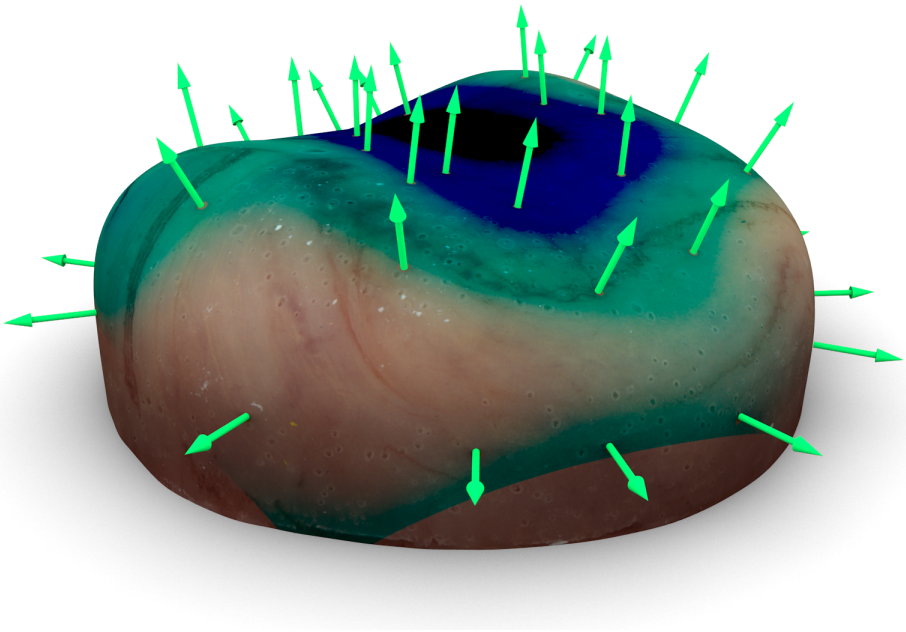
9.7 Contribution

The two key contributions of this chapter are:

- Unifying previousy registration methods in literature under a common framework.
- Predicting improvements to existing registration methods using the framework.

Part V

Application and systems development



Chapter 10

Real-time Augmented Reality System for Improving Situational Awareness

10.1 Introduction

Robot-assisted minimally invasive surgeries are becoming increasingly popular as they provide increased dexterity and control to the surgeon while also reducing trauma, blood loss and hospital stays for the patient [248]. These devices are typically tele-operated by the surgeons using visual feedback from stereo-cameras, but without any haptic feedback. This can result in the surgeon relying only on vision to identify tumors by mentally forming the correspondence between intra-operative view and pre-operative images such as CT scans/MRI, which can be cognitively demanding.

Automation of simple but laborious surgical sub-tasks and presenting critical information back to the surgeon in an intuitive manner has the potential to reduce the cognitive overloading and mental fatigue of surgeons [249]. This work leverages the recent advances in force sensing technologies [8], tumor localization strategies [2, 137, 138], online registration techniques (see Chapter 4, 6) and augmented reality [250] to automate the task of tumor localization and dynamically overlay the

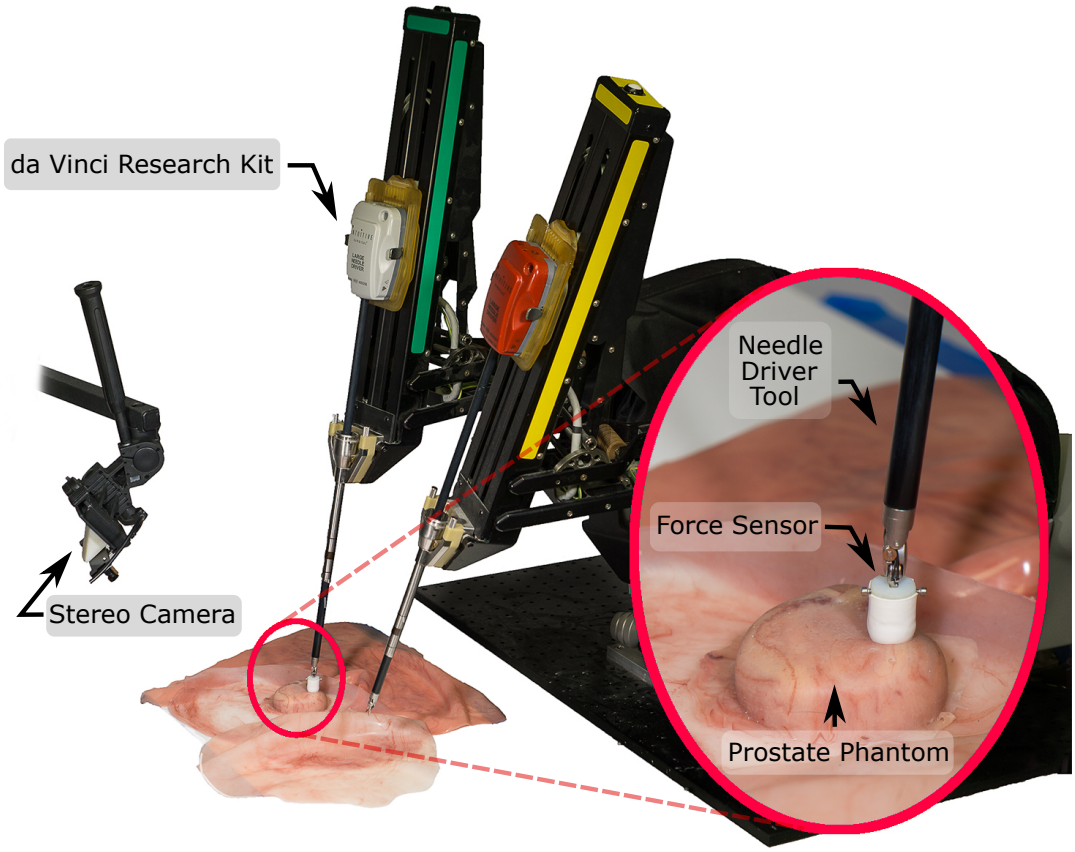


Figure 10-1: Experimental setup showing the dVRK robot with a miniature force sensor attached to the end-effector. A stereo camera overlooks the workspace of the robot. A phantom prostate with embedded stiff inclusion is placed in the workspace of the robot.

information on top of intraoperative view of the anatomy.

While the works in literature deal with force sensing [251, 252], tumor localization [2, 249, 137, 111, 138, 112] and graphical image overlays [253, 254, 255, 256], there is a gap in literature when it comes to systems that deal with all these issues at the same time. For example, Yamamoto *et al.* [256] deal with tumor localization and visual overlay, but they assume the organ is flat and place the organ on a force sensing plate, which is not representative of a surgical scenario. On the other hand, Garg *et. al.* [249] use a palpation probe mounted on a da Vinci research kit (dVRK) tool (This probe was originally developed by McKinley *et. al.* [257]). However, they do not deal with registering the organ or visual overlay of the estimated stiffness map.

The work of Puangmali *et. al.* [251] (and many others as noted by Tiwana *et. al.* [252] in their review paper) have developed miniature force sensing technologies for minimally invasive surgeries, but they do not discuss strategies to palpate or ways to perform image overlay of the estimated stiffness map. The prior work from our group including those of Ayvali *et. al.* [2, 137] and Salman *et. al.* [138] develop techniques to smartly palpate a tissue and search for tumors with a small number of probings. The work of our collaborators Chalasani *et. al.* [111, 112] provide an alternate way of smart palpation by probing along continuous trajectories. However, these works do not deal with graphically overlaying the estimated stiffness map. These works also often use an ad hoc force sensing set up consisting of a plate mounted on a commercial force sensor. Finally the work of Wang *et. al.* [258, 114] uses force controlled exploration to perform registration to preoperative model of the organ, but they do not estimate the stiffness of the tissue or perform graphical image overlay. The work in this chapter aims to bridge these shortcomings and present a unified system capable of addressing all the above mentioned issues at the same time.

The system of Naidu *et. al.* [259] comes closest to our work. They use a custom designed tactile probe (developed by Trejos *et. al.* [260]) to find tumors and visually overlay the tactile image along with the ultrasound images. The wide tactile array that they use, allows for imaging sections of the organ instead of obtaining discrete measurements, as in our case. This eliminates their need to develop sophisticated tumor search algorithms. However, as acknowledged by Trejos *et. al.* [260], it is not clear as to how their system would perform when using non-flat organs such as prostates and kidneys; since the tactile array cannot deform and confirm to the shape of the organ. Without performing registration, the image overlay would also be affected on non-flat organs.

The framework presented in this work is robot agnostic and modular in nature. We demonstrate the efficacy of the system by performing autonomous tumor localization on a phantom prostate model and other custom fabricated flat silicone organs with embedded tumors using the dVRK platform [134] (see Fig. 10-1). There are two reasons for choosing the dVRK for demonstration – (1) it is a good representation

of a surgical robot, (2) there are several research groups across the world that use dVRK and we hope they will all benefit from the work presented in this chapter. A miniature force sensor mounted at the tip of the dVRK needle driver tool (developed by Li *et. al.* [8]) is used to sense the contact forces. An active tumor search strategy developed by Salman *et. al.* [138] is used to localize the tumor. The estimated stiffness map is overlaid on a registered model of the anatomy and displayed in real-time on a stereo viewer. The overall focus of this chapter is to combine the various contributions of this thesis into a demonstrative surgical system that would enable fast estimation of registration, tumor search and graphical image overlay. We believe that our contributions would be used in the software framework being developed by our collaborators Chalasani *et. al.* [130] at Johns Hopkins university and Vanderbilt university to provide online complementary situational awareness (CSA) for surgical assistance.

10.2 Related Work

10.2.1 Tumor search approaches

The recent developments in force sensors have also resulted in a number of works that automate mapping of the surface of the anatomy to reveal stiff inclusions. The different palpation strategies commonly used are: discrete probing motion [256, 261], rolling motion [110], cycloidal motion [262] and sinusoidal motion [111, 112]. Some of these works direct the robot along a predefined path that scans the region of interest on the organ [263, 256, 113], while others adaptively change the grid resolution to increase palpation resolution around boundaries of regions with high stiffness gradients [262, 261].

Over the last two years, Bayesian optimization-based methods have gained popularity [2, 249, 137, 112]. These methods model the stiffness map using a Gaussian process regression (GPR) and reduce the exploration time by directing the robot to stiff regions. While the objective of most prior works is to find the high stiffness

regions [2, 249, 137], recent work by Salman *et. al.* [138] on active search explicitly encodes finding the location as well as the shape of the tumor as its objective.

10.2.2 Surgical registration and image overlay

There is a rich literature of image overlay for minimally invasive surgeries [253], including some works on augmented reality in human-surgeries [264]. Often the image that is overlaid is a segmented preoperative model, and it manually placed in the intraoperative view [264, 255]. Very few works such as [254, 265], deal with manual placement followed by automatic registration of the organ models. There are a number of registration techniques that have been developed for surgical applications; the most popular one being ICP [11] and its variants (See Rusinkiewicz *et. al.* [12] for a survey on variants of ICP). There also exist methods that deal with local deformation caused by tool-tissue interaction, such as the CMU approach in Chapter 3, and global deformation caused by organ shift, swelling, etc. [116, 258, 114].

Probabilistic methods for registration have recently gained attention as they are better at handling noise in the measurements. Billings *et al.* [23] use a probabilistic matching criteria for registration, while methods such as [5, 6] (see Chapter 6, 9 for more details) use Kalman filters to estimate the registration parameters. In Chapter 6 we reformulate registration as a linear problem in the space of dual quaternions and uses a Bingham filter and a Kalman filter to estimate the rotation and translation respectively. Such an approach has been shown to produce more accurate and fast online updates of the registration parameters.

While the above literature deals with registering preoperative models onto an intraoperative scene, there is very little literature that deals with overlaying stiffness maps on the preoperative models and updating the maps in real-time as new force sensing information is obtained. Real-time updates are very important because they give the surgeon a better sense of what the robot has found and gives them insight into when to stop the search algorithm which is a subjective decision, as observed by Ayvali *et. al.* [137]. The works of Yamamoto *et al.* [256] and Naidu *et al.* [259] are exceptions and deal with dynamic overlaying of the stiffness image, but only onto

pre-registered flat organs. Their approaches do not generalize to the cases of non-flat organs such as kidneys or prostates that we consider in this work.

10.2.3 Force sensing for surgical applications

Several survey papers report a number of devices that measure contact forces [251, 252]. Some common drawbacks with many existing devices are: difficulty to sterilize, high cost, delicate components and lack of flexibility of form factor. Recently, Li *et al.* [8] have developed a miniature force sensor that uses an array of thin-film force sensitive resistor (FSR) with embedded signal processing circuits. A diagram of the sensor can be seen in Fig. 10-2. This sensor is light-weight, inexpensive, robust, and has a flexible form factor.

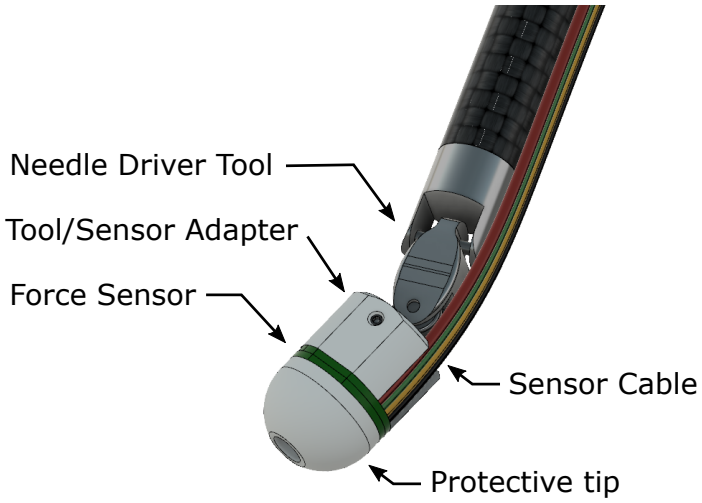


Figure 10-2: The FSR sensor that we use in this chapter was developed by Li *et al.* [8]

10.3 Problem Setting and Assumptions

We use an ELP stereo camera (model 1MP2CAM001) overlooking the workspace of a dVRK [134]. A custom fabricated prostate phantom (made using Ecoflex 00-10) as well as two other flat silicone organs, all embedded with plastic pieces to mimic stiff tumors, are used for experimental validation.

Given an *a priori* geometric model of an organ, the measurements of the tool tip positions and associated contact forces, and stereo-camera images of the intraoperative scene, our goal is to (i) register the camera-frame, robot-frame and model-frame to each other, (ii) estimate the stiffness distribution over the organ's surface, and (iii) overlay the estimated stiffness distribution on the registered model of the organ and display it back to the user.

We make the following assumptions in this work:

- The shape of the organ deforms only locally due to tool-interaction.
- The tool-tip pose can be obtained accurately from the robot kinematics.
- The forces applied by the tool are within the admissible range (less than 10N) in which the organ only undergoes a small deformation (less than 8mm) that allows it to realize its undeformed state when the force is removed.
- The stiff inclusion is located relatively close to the tissue surface, so that it can be detected by palpation.

10.4 System Modeling

Fig. 10-3 shows the flowchart of the entire system. Modules such as camera calibration, model generation and registration need to be run only once at the beginning of the experiment. On the other hand, the tumor search, probing, and augmented display modules are run in a loop until the user is satisfied with the result and halts the process. While the system is largely autonomous, user input is required in two steps: (i) Camera-model registration, to select the organ of interest in the view of the camera, (ii) selecting region of interest for stiffness mapping. The modularity of the system allows the user to choose any implementation for registration, force-sensing and tumor localization. The important modules of our system are discussed in detail in the following sections.

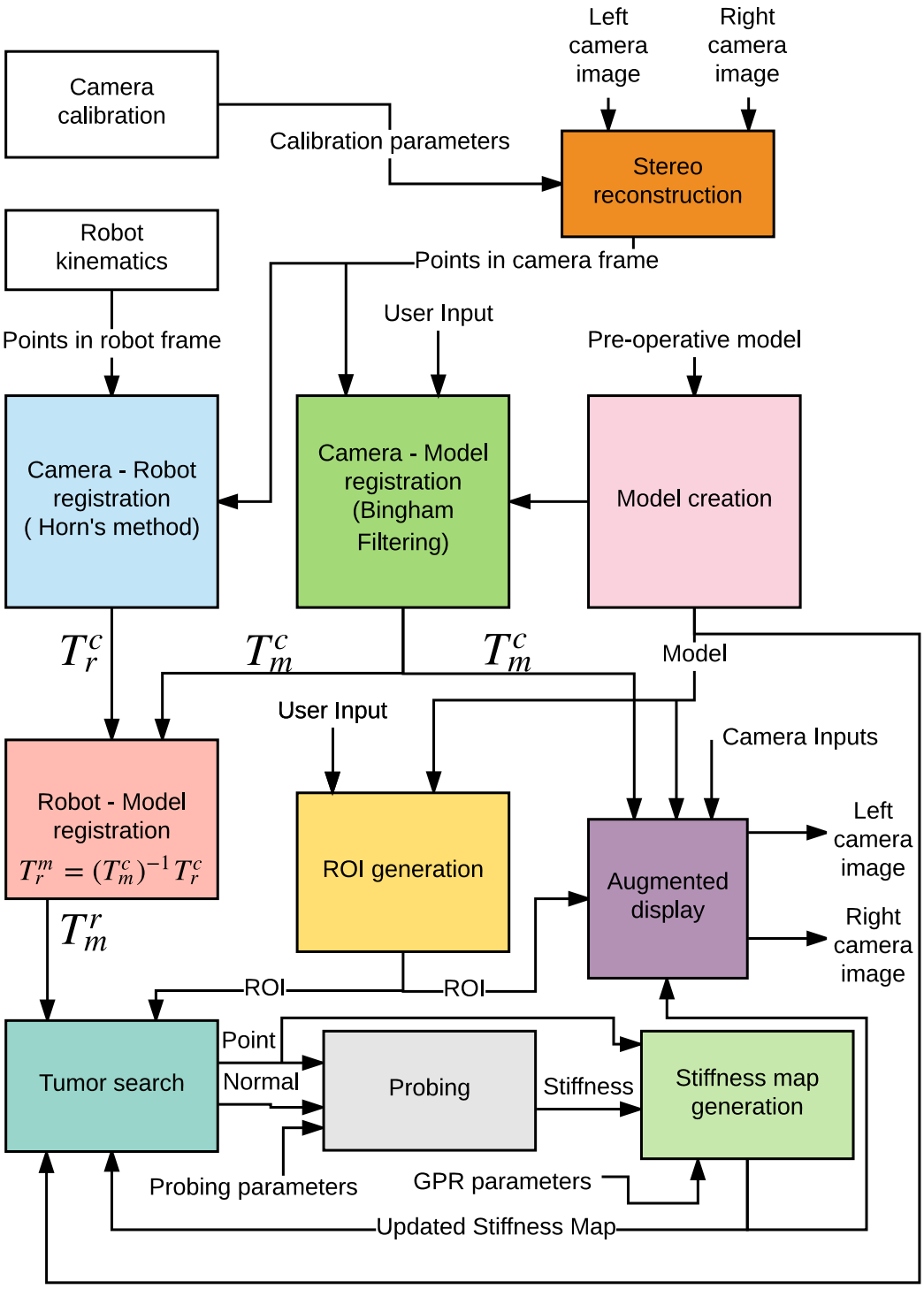


Figure 10-3: Flowchart showing all the modular components of our system. Some of the modules such as camera calibration, stereo reconstruction, model creation, and camera-robot-model registrations are performed once before the start of the experiment, while the other modules are constantly run for the duration of the experiment.

10.4.1 Registering Camera and Robot Frames

The cameras are calibrated using standard robot operating system (ROS) calibration [133]. The robot is fitted with a colored bead on its end effector that can be easily segmented from the background by hue, saturation, and value. Registration between the camera-frame and the robot-frame is performed by the user through a graphical user interface (GUI) that shows the left and right camera images and has sliders representing color segmentation parameters.

The robot is moved to a fixed set of six points and the tip positions are recorded from the kinematics. These points are chosen to cover a substantial amount of the robot's workspace, stay within the field of view in the camera, and not contain symmetries that would make registration difficult. We chose to use only six points after experiments showed that additional points failed to significantly decrease the root mean squared (RMS) error, as shown in Table 10.1. For each of the points, we perform a series of actions.

Table 10.1: RMS error vs number of points used

Number of points	5	6	7	8	11	51
RMS error (mm)	2.71	2.37	2.84	3.01	2.82	2.85

First, we move the robot to the specified location, then we process both the left and right images to find the centroid of the colored bead fitted to the robot. The centroid of the ball in pixels is found as the center of the minimum enclosing circle of the contour with the largest area. We repeat this for as many frames as are received over ROS in one second (in our case 15), and the centroid is then averaged over all frames to reduce the effect of noise in the image. The centroid is drawn onto both images in the GUI, allowing the user to evaluate the accuracy of the centroid estimation. The pixel disparity is calculated as the difference between the x coordinates of the centroid in the left and right images. This disparity is fed into a stereo-camera model that ROS provides, to calculate a 3D point in the camera-frame.

Following this, we obtain six points in both the camera-frame and the robot-frame (using the kinematics of the robot). We use Horn's method [4] to calculate the

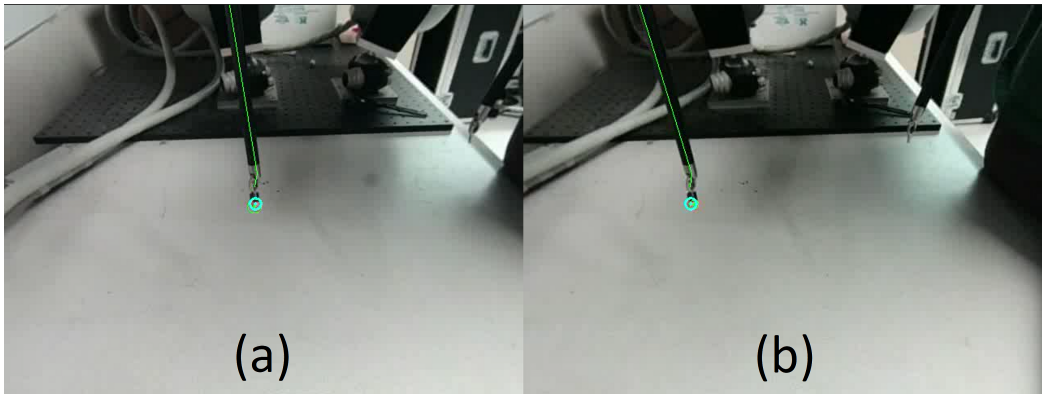


Figure 10-4: (a) and (b) are left and right camera views showing the dVRK robot tool. Cyan colored circles are drawn on both the images by finding the tool tip position from the robot kinematics and then using the registration between robot frame and camera frame. Note how the position of the circles visually confirms well with the true location of the tool tip.

transformation $\boldsymbol{T}_m^c$ between the camera and the robot frames. This transformation is saved to a file and the calculated RMS error is displayed to the user. In addition, the robot's current position is transformed by the inverse of the calculated transformation and projected back into the pixel space of both cameras. Circles are drawn at these pixel positions in the left and right images in the GUI so that the user can visually confirm that the registration is successful and accurate (see Fig. 10-4).

10.4.2 Registering Camera and Preoperative Model Frames

The transformation between camera-frame and model-frame, $\boldsymbol{T}_m^c$ is estimated by registering the reconstructed point cloud from stereo images with the preoperative model of the organ. The intraoperative scene as viewed by the stereo cameras is as shown in the top of Fig. 10-5. A user manually selects the region containing the organ of interest. Following this the user can also further refine the selection using a graph cut-based image segmentation (developed by Greig *et. al.* [266]).

A Bingham distribution-based filtering approach is used to automatically register the stereo point cloud to the preoperative model (see Chapter 6). The mean time taken to register is 2s and the RMS error is 1.5mm. The center row in Fig. 10-5 shows the registered model of the organ overlaid on the stereo views. Note how the pose of

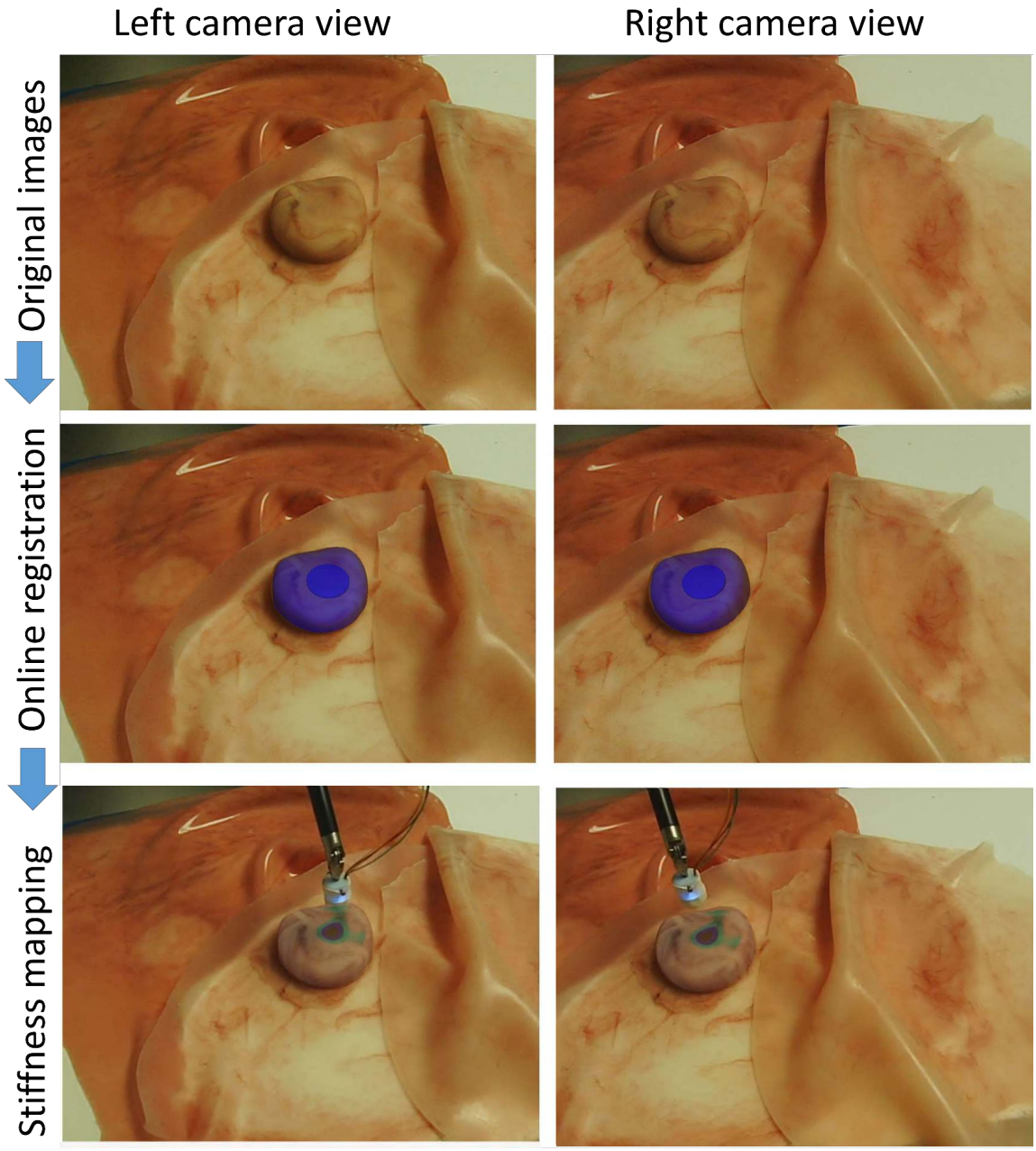


Figure 10-5: Top row: Original left and right camera images. Middle row: Camera images with registered prostate model shown in semi-transparent blue. The tumor model is also shown to allow us to compare our stiffness mapping result. Bottom row: The robot probes the organ and records force-displacement measurements. The estimated stiffness map is then augmented on the registered model in this figure. Dark blue regions show high stiffness. Note that the stiffness map reveals the location and shape of the tumor.

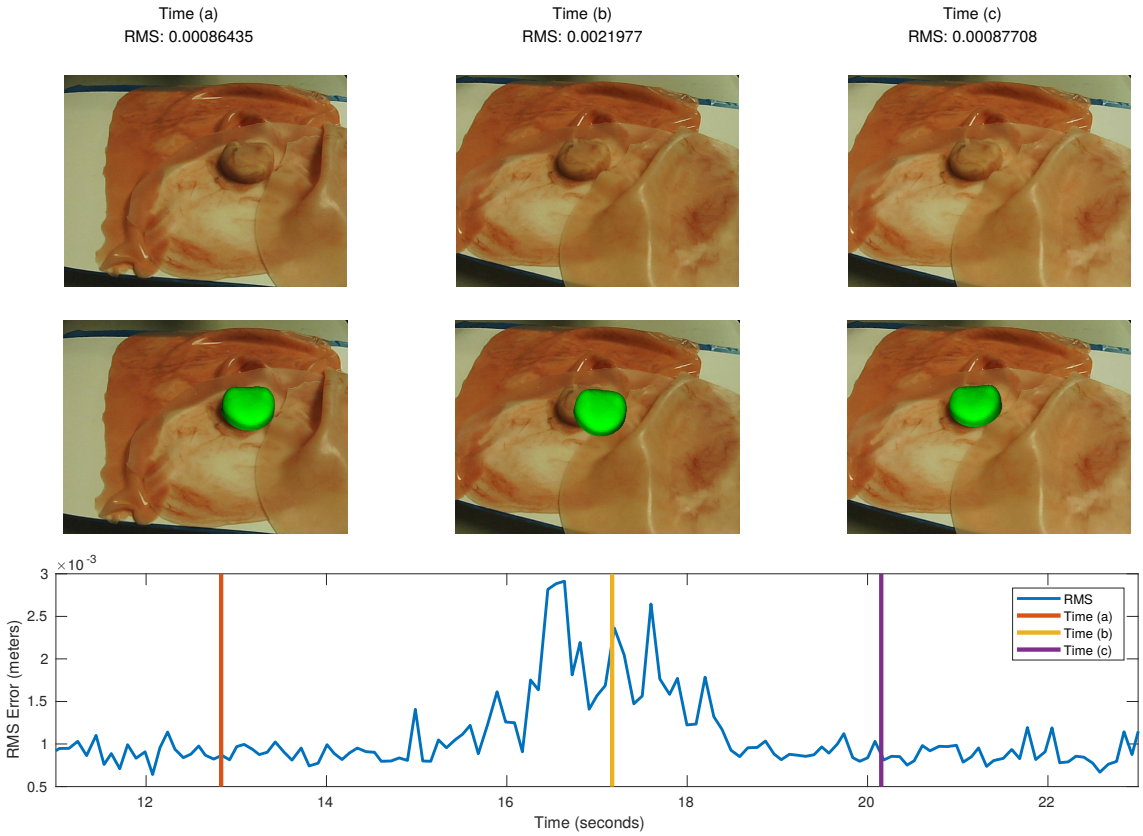


Figure 10-6: Registration results on a moving phantom prostate shown at three different times. Note the large disturbance at time (b) caused by applying an external force to the organ and the recovered tracking at time (c).

the registered model accurately matches the pose of the organ. In the same figure we also show the model of the tumor in the registered view to highlight how accurately the stiffness map estimates the location of the tumor (see bottom row of Fig. 10-5)

We also augmented the static graph-cut based segmentation using an automatic traveling mask. Once we are satisfied that the model is roughly registered to the object, we switch from the static graph-cut mask to the traveling mask. Using the same rendering engine used in the GUI, a z-depth buffer is rendered from the camera's view of the model. This depth buffer is scaled from 0 to 255 with 0 representing the pixel farthest from the camera and 255 representing the closest. Using this buffer, we create a new mask for our camera image by masking out all pixels with a depth of zero, effectively creating a cutout of our rendered model. Because we render the depth buffer every time the model's estimated transformation is changed, we create an image mask that moves along with our model. Results using this traveling mask can be seen in Fig. 10-6. Tracking is maintained even after we apply a large disturbance to move the silicone prostate out from under the tracked model. Although extreme movements will certainly make our system lose tracking, the system is robust to small periodic movements similar to what might be seen in the body, due to physiological motions such as respiration, heart beat, etc. as well as due to interactive forces during palpation. To the best of our knowledge, the prior works on palpation-based stiffness mapping assume the organ is rigidly clamped with respect to the robot and ignore movement of the organ altogether [112, 259, 113, 109].

10.4.3 Tumor Search and Stiffness Mapping

The problem of tumor search is often posed as a problem of stiffness mapping, where the stiffness of each point on a certain organ is estimated, and regions with stiffness higher than a certain threshold are considered as regions of interest (tumors, arteries, etc.). The framework that we use for localizing tumors utilizes Gaussian process (GP) to model the stiffness distribution combined with a GP-based acquisition function to direct where to sample next for efficient and fast tumor localization. This is described in Fig. 10-7.

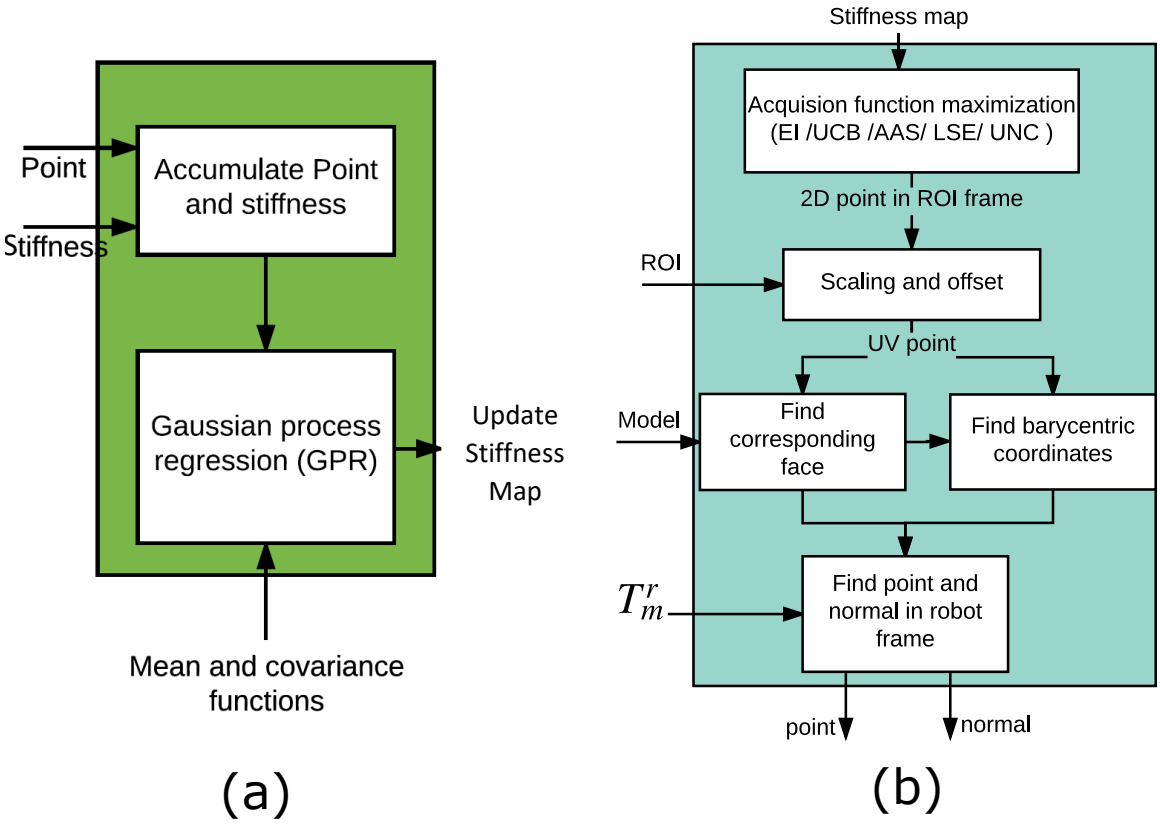


Figure 10-7: (a) An expanded flowchart of the "Stiffness map generation" block shown in Fig. 10-3. As the robot probes a new point on the organ, it calculates the stiffness value at that point. All the probed points with their corresponding stiffness values are accumulated and used to train a GPR (with predefined mean and covariance functions) which is used to updates the estimated stiffness map. (b) An expanded flowchart of the "Tumor search" block shown in Fig. 10-3

By using GP, we assume a smooth change in the stiffness distribution across the organ. Since every point on the organ's surface can be uniquely mapped to a 2D grid, the domain of search used is $X \subset \mathbb{R}^2$. The measured force and position after probing the organ by the robot at $\boldsymbol{x}$ provides the stiffness estimation represented by y.

The problem of finding the location and shape of the stiff inclusions can be modeled as an optimization problem. However, an exact functional form for such an optimization is not available in reality. Hence, we maintain a probabilistic belief about the stiffness distribution and define a so called "acquisition function", ξ_{acq}, to determine where to sample next. This acquisition function can be specified in various ways and thus our framework is flexible in terms of the choice of this acquisition function that is being optimized. Prior works have considered various choices for the acquisition functions such as expectation improvement (EI) [2, 137], upper confidence bound (UCB) [249], uncertainty sampling (UNC), active area search (AAS) and active level set estimation (ALSE) [138].

While our system is flexible to the choice of acquisition function, in this work we demonstrate tumor localization using active level set estimation (ALSE). ALSE determines the set of points, for which an unknown function (stiffness map in our case) takes value above or below some given threshold level h. The mean and covariance of the GP can be used to define a confidence interval,

$$Q_t(\boldsymbol{x}) = \left[\mu_t(\boldsymbol{x}) \pm \beta^{1/2}\sigma_t(\boldsymbol{x})\right] \tag{10.1}$$

for each point $\boldsymbol{x} \in \bar{X}$. Furthermore, a confidence region C_t which results from intersecting successive confidence intervals can be defined as,

$$C_t(\boldsymbol{x}) = \bigcap_{i=1}^{t} Q_i(\boldsymbol{x}). \tag{10.2}$$

ALSE then defines a measure of classification ambiguity $a_t(\boldsymbol{x})$ defined as,

$$a_t(\boldsymbol{x}) = \min\left\{\max(C_t(\boldsymbol{x})) - h, h - \min(C_t(\boldsymbol{x}))\right\}. \tag{10.3}$$

ALSE chooses sequentially queries (probes) at $\boldsymbol{x}_*$ such that,

$$\boldsymbol{x}_* = \underset{\boldsymbol{x} \in X}{\operatorname{argmax}}\, a_t(\boldsymbol{x}). \tag{10.4}$$

For details on how to select the parameter h, we refer the reader to the work of Gotovos *et. al.* [267].

10.4.4 Probing and Force Sensing

We affixed a miniaturized Tri-axial sensor developed by Li *et. al.* [8] onto the needle driver tool for the dVRK to provide contact force measurements (see Fig. 10-1). The force sensor is a force sensitive resistor (FSR) based force-to-voltage transducer operating in thru-mode electrodes configuration. The design combines FSR array with a center mounted pre-load mechanical structure to provide a highly responsive measurement of contacting force and direction of the force vector. In this experiment, we electrically bridged the four sensing array elements on the force sensor, to provide improved sensitive force measurement along the normal direction of the sensor, since the dVRK can be accurately oriented to probe along the local surface normal. In addition, we implemented online signal processing software in the sensor embedded controller, for analog signal amplification, filtering, automatic self-calibration, which is crucial step to improve sensor performance when using inexpensive force sensing materials such as 3M Velostat film from Adafruit.

First, the robot is commanded to a safe position $\boldsymbol{p}_1$ which is at a known safe height z_{safe} as shown in Fig. 10-8(b). The robot is then commanded to move to position $\boldsymbol{p}_2$ which is at an estimated distance λ from the desired probing point $\boldsymbol{p}_0$, along the normal to the surface at $\boldsymbol{p}_0$, $\boldsymbol{n}$ (see Fig. 10-8(b)). While maintaining its orientation, the tool is commanded to move to position $\boldsymbol{p}_3 = \boldsymbol{p}_2 - (\lambda + d_{max})\boldsymbol{n}$. The force and position data are constantly recorded as the robot moves from $\boldsymbol{p}_2$ to $\boldsymbol{p}_3$. When the force sensor contacts the tissue surface, if the contact force exceeds a set threshold F_{max} or if the probe penetrates more than a set depth d_{max}, the robot is no longer moved. This ensures that the probing does not hurt the patient or cause any damage

to the robot. Following this we retract the robot to position $\boldsymbol{p}_2$ and then $\boldsymbol{p}_1$. Note that we do not record force and displacement data during the retraction process.

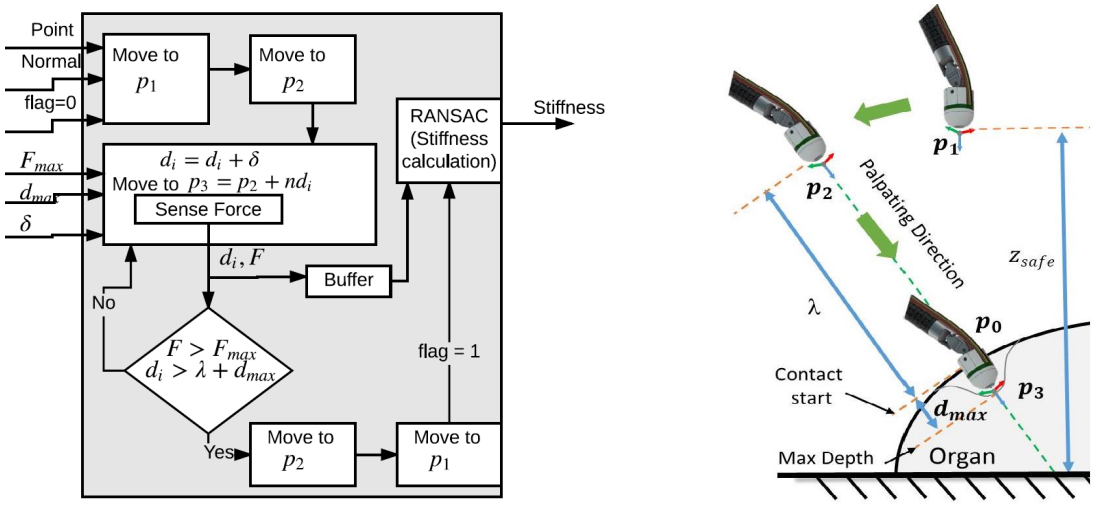


Figure 10-8: (a) An expanded flowchart of the "Probing" block shown in Fig. 10-3. (b) The various steps taken to probe a desired point along a desired normal direction as provided by the tumor search module.

Next the recorded data is treated as input to the stiffness mapping algorithm similar to the complementary model update described in Chapter 3. There are two important steps of this algorithm: (i) baseline removal, (ii) stiffness calculation. Ideally, the force sensor reading should be zero when there is no contact between force sensor and the interest area. However, in reality there is always a small residue in the sensor readings even when there is no contact. Thus we find the mean sensor output value when the probe is at $\boldsymbol{p}_2$ and then subtract all the subsequent measurements from this baseline force. For stiffness calculation, we use a standard RANSAC algorithm to find the best fit line between the y-axis (force sensor data) and x-axis (displacement data). As a result, the calculated regression coefficient indicates the changing rate of the contact force respect to a unit displacement, which can be used as the best approximation of stiffness value. Fig. 10-9 shows the nearly linear variation of force with displacement, justifying the use of slope of the best fit line as an approximation for the stiffness.

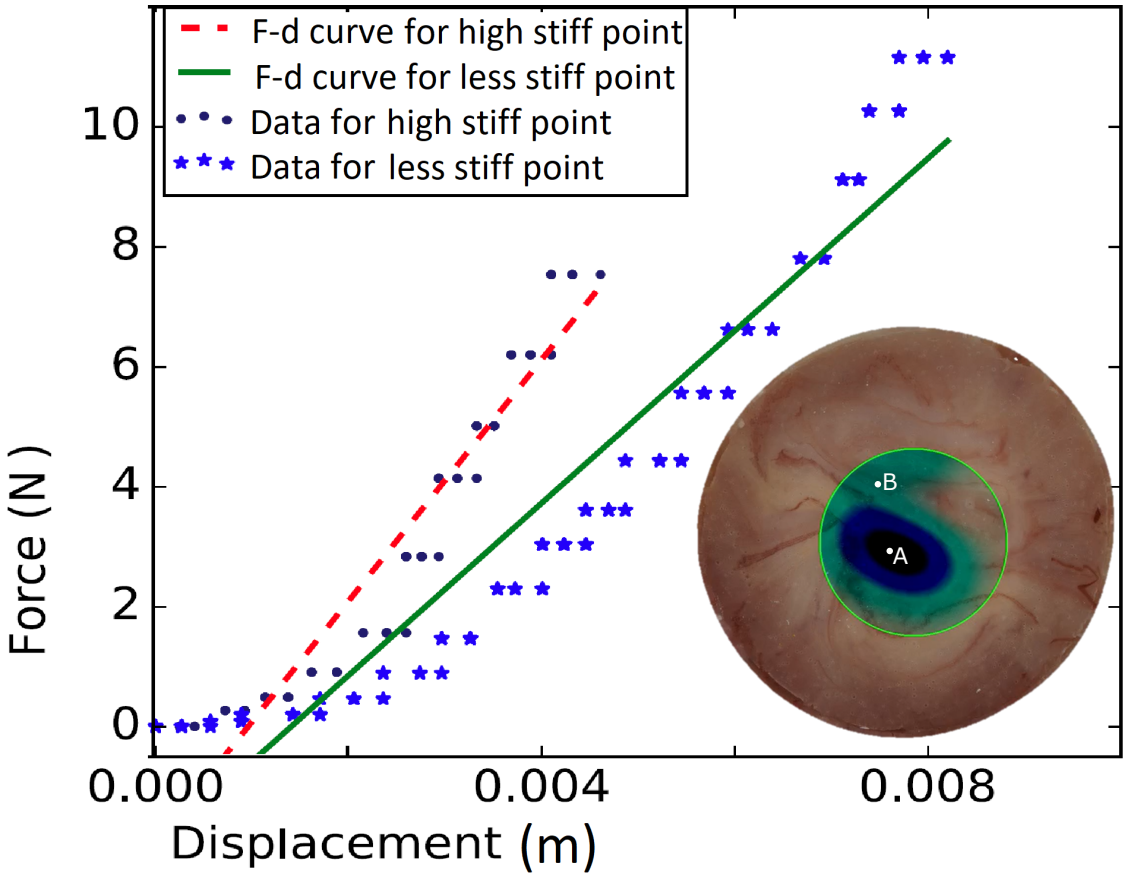


Figure 10-9: The plot shows forces vs displacement for two sample points A and B on the surface of the organ. Note that the forces are limited to 10N and the displacement is also restricted to 8mm. RANSAC is used to find the best-fit line and the slope gives us an estimate of the stiffness at the probed location. The circular 2D space forms a one-to-one mapping with the 3D surface of the organ. The green circle represents the user-defined ROI. The stiffness map is estimated in this ROI. Different shades of blue are used to represent the stiffness values. Point A is located in on a stiff region, while B is located on a soft region. The plot reveals the corresponding stiffness.

10.4.5 Dynamic Image Overlay

The rendering of the overlays is done using the Visualization Toolkit (VTK). Two virtual 3D cameras are created to match the real cameras using the results of camera calibration. The pre-operative model is placed in virtual 3D space according to the camera-to-organ registration, $\boldsymbol{T}_m^c$, and rendered as a polygonal mesh from the perspective of each camera. These two renders are overlaid onto live video from the left and right camera feeds as their backgrounds. These renderings are displayed in a GUI divided into three tabs. The first tab is for registration, which overlays the pre-operative model as described above and additionally allows the user to mask and segment the point cloud as described in Sec. 10.4.2. It also provides buttons to start and stop model registration. The second tab allows the user to select a ROI defined in a 2D UV texture map that represents a correspondence between pixels on a 2D image to 3D coordinates on the surface of the preoperative model (see Fig. 10-9(c)). The third tab overlays the pre-operative model over the camera feeds and allows the user to set the opacity of the overlay using a slider at the bottom of the window.

In addition, the renderings in the third tab add a texture to the rendered model. For this texture, the results of the tumor search are turned into a heat-map image representing relative stiffness in a user-specified ROI (see Fig. 10-9(c)). This ROI is defined in 2D UV texture coordinates that represent a correspondence between pixels on a 2D image to 3D coordinates on the surface of the polygonal mesh. The heat-map image is broadcast over ROS and overlaid onto the pre-operative model's 2D texture image resulting in dark marks in high-stiffness areas while preserving texture details found in the preoperative model's original texture (see Fig. 10-9(c)). This 2D texture is then applied to the polygonal mesh using the UV map, resulting in a 3D overlay of the stiffness map onto the video feed from each camera. Fig. 10-10 shows the stiffness maps at various stages of probing, dynamically overlaid on the registered model of the organ. Note that the stiffness map clearly reveals the location and shape of the tumor which is shown in the middle row of Fig. 10-5.

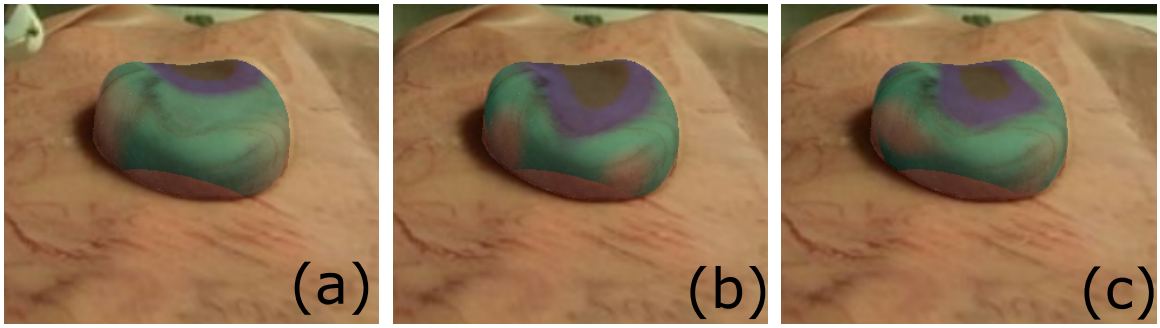


Figure 10-10: The figures show the augmented stiffness map at various stages of probing. The high stiffness regions are shown in darker shades of blue, while the low stiffness regions are in lighter shades of blue.(a) Result after a single probe, (b) result after 4 probings, (c) result after 10 probings.

10.5 Experiments and Validation

In this section, we validate our system through experimentation on various phantom models. In addition to the experiments performed on the silicone prostate phantom, an experiment was performed on a custom fabricated flat silicone organ embedded with plastic inclusions to mimic stiff tumors. We used the dVRK robot with the organ placed on a force sensing plate fitted with a commercial force sensor created by ATI to generate the ground truth stiffness maps.

Upon generating the ground truth stiffness maps using a raster scan pattern with a high density of probed points, the silicone organs were registered and probed using the registration and search methods described in Sec 10.4. It is worth noting, that unlike Yamamoto *et. al.* [256], we do not assume the flat organ is pre-registered. We estimate the registration from the stereo-camera and use the estimated registration for the overlay. Also the force sensing plate is only for ground truth stiffness mapping. For the actual experiments, we use the miniature force sensor shown in Fig. 10-2. The resulting stiffness maps, as well as a comparison of how the maps appear when overlaid in our GUI, can be seen in Fig. 10-11 and 10-12. These figures show that our system is able to capture the position and size of the tumors with far fewer probed points in a fraction of the time taken to generate the dense stiffness map[1]. Fig. 10-13(a) and

[1]The experiments with our system took a total of 3 minutes to execute as opposed to the raster scan that took upwards of 90 minutes.

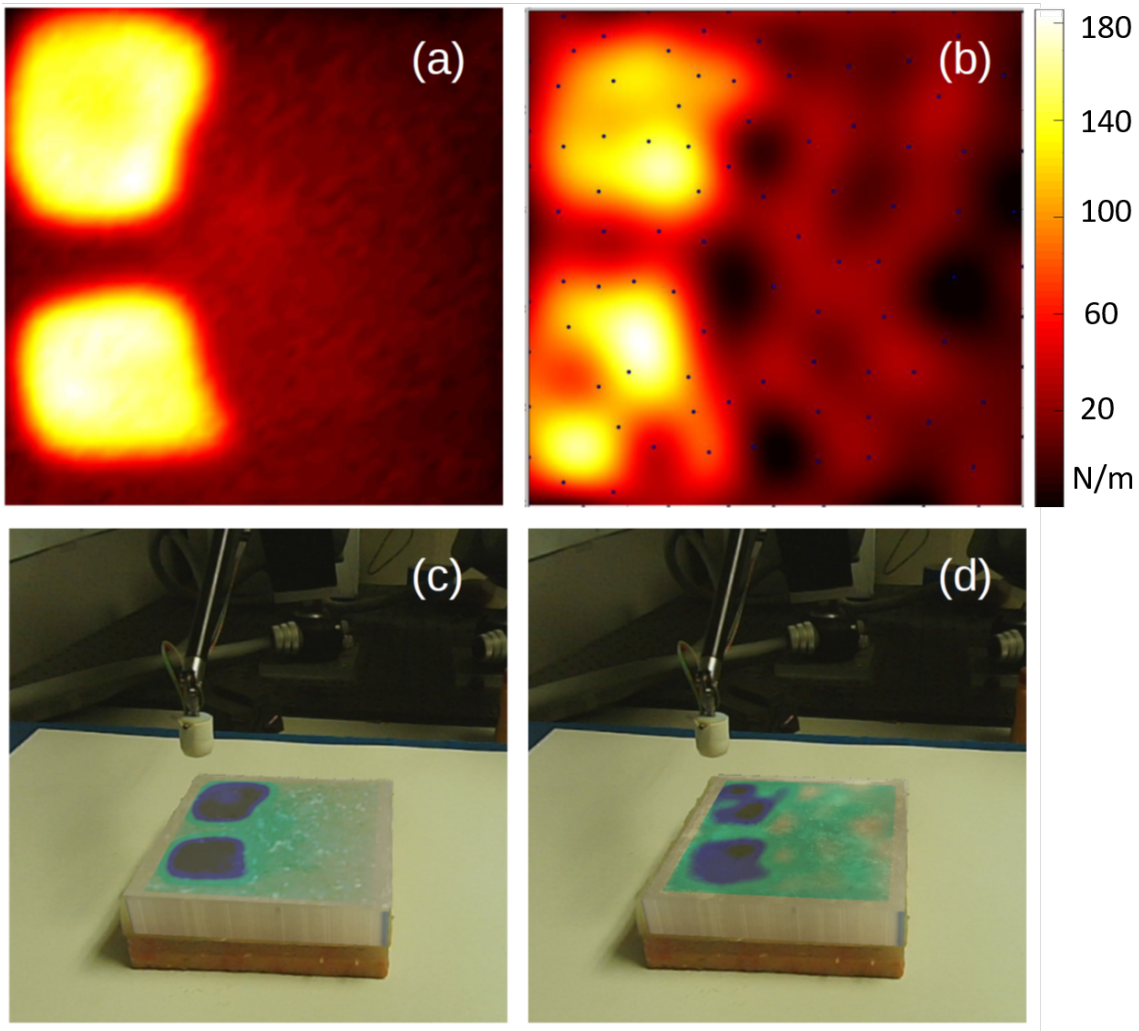


Figure 10-11: Experiments on the flat silicone organ. (a) The ground truth stiffness map. (b) The stiffness map as estimated by our system. The probed locations are shown by blue dots. (c) The ground truth stiffness map overlaid on top of the phantom organ. (d) The estimated stiffness map overlaid on top of the phantom organ. The stiffness maps are shown in the space of the UV map. The x and y range for ground truth and estimated stiffness maps are the same.

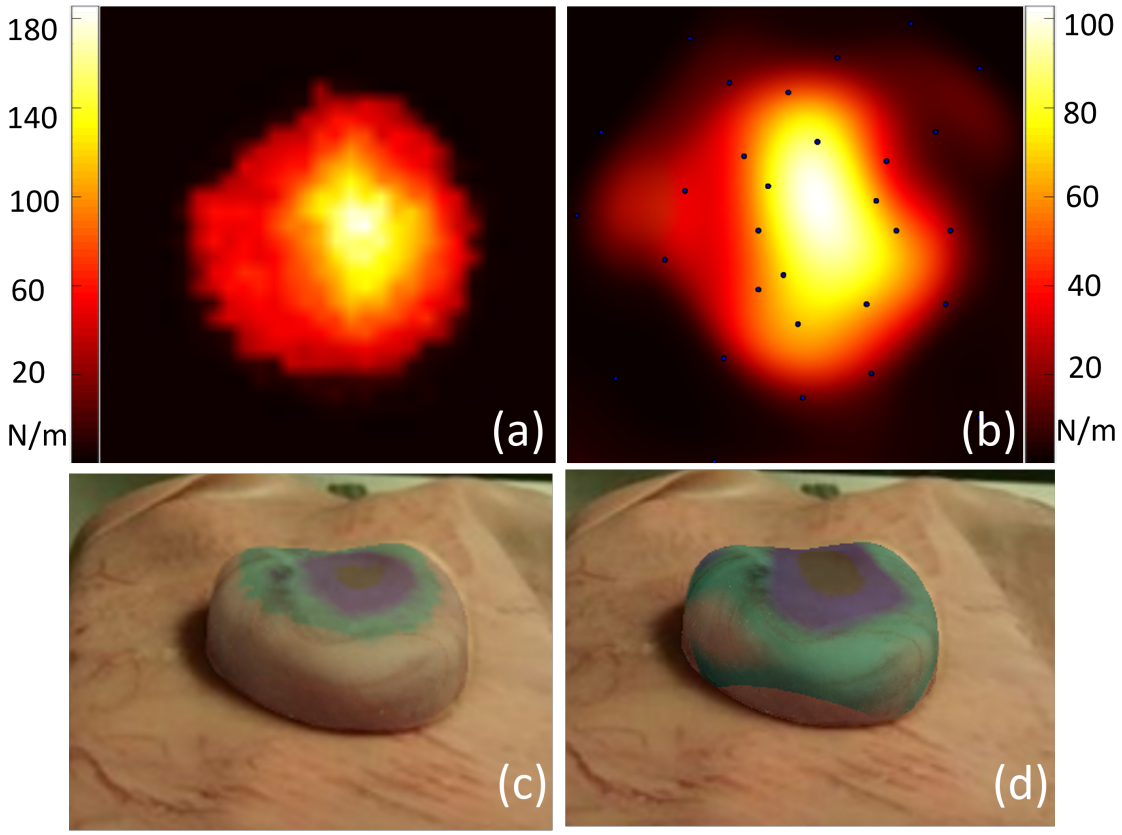


Figure 10-12: Experiments on the phantom prostate. (a) The ground truth stiffness map. (b) The stiffness map as estimated by our system. The probed locations are shown by blue dots. (c) The ground truth stiffness map overlaid on top of the phantom organ. The stiffness maps are shown in the space of the UV map. The x and y range for ground truth and estimated stiffness maps are the same. (d) The estimated stiffness map overlaid on top of the phantom organ.

Table 10.2: Accuracy, recall and precision of estimated stiffness maps

	Accuracy (%)	Recall (%)	Precision (%)
Prostate	98.08	82.04	85.77
Flat organ	88.50	92.11	72.21

(b) show the front and back view of the phantom prostate with the overlaid image of the estimated stiffness along with the surface normals at the locations the organ was probed. As is evidenced by this figure, our system is capable of probing and overlaying the image on a non-flat highly curved organ.

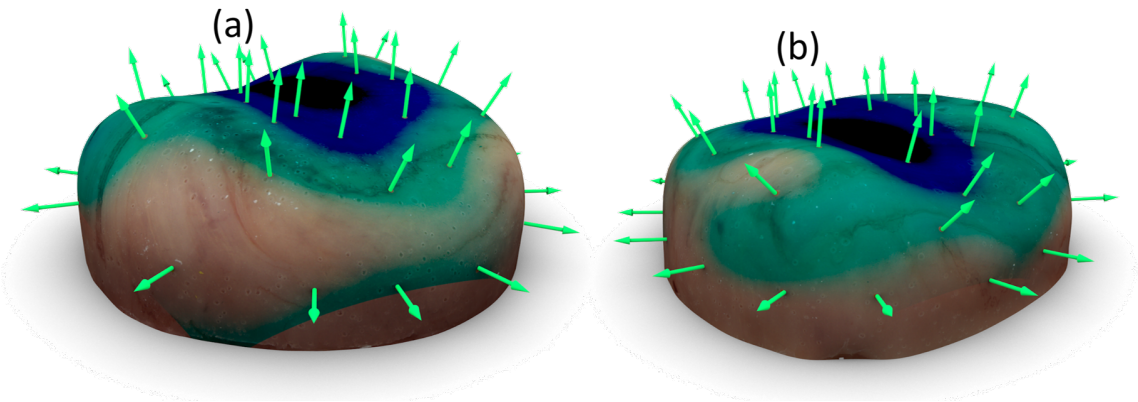


Figure 10-13: The prostate model with overlaid stiffness map. The surface normals at the various probed locations are shown by the green arrows. (a) Front view (b) Back view.

Table 10.2 shows the precision, accuracy and recall of the estimated stiffness maps when compared to the ground truth. Precision, accuracy and recall are popular metrics to compare performance of regions-of-interest detection problems and have been used to compare stiffness maps in literature [138]. The results show that the shape and location of tumor as estimated by our approach is accurate and closely matches the ground truth. Although the exact shape of the tumors is not perfectly captured for the case of the flat organ, the resulting map is more than enough to show the user where the tumors are located. The RMS error in the stiffness estimation for the prostate phantom is 18.71 N/m and for the flat organ is 40.09 N/m, which is sufficient for a surgeon to differentiate tumor from tissue as noted by Chalasani *et. al.* [111, 112].

10.6 Discussions and Future Work

In this chapter, we presented a system that unifies autonomous tumor search with augmented reality to quickly reveal the shape and location of the tumors while visually overlaying that information on the real organ. Our system is capable of probing highly curved organs as well as tracking the movement of the organ that may be caused by the forceful interaction with the tissue. This has the potential to reduce the cognitive overload of the surgeons and assist them during the surgery. Our system demonstrates promising results in experimentation on phantom silicone organs.

While we demonstrate the task of stiffness mapping in this chapter, our system can be used to visually overlay pre-surgical plans, ablation paths, annotate important landmarks, etc. to aid the surgeon during the procedure. In our future work we plan to account for large deformations of the organ and update the model accordingly. We plan to utilize computationally fast approaches to segment the dVRK tools from the images and avoid any obstructions to the overlaid stiffness map. Furthermore, as demonstrated by other researchers in this field, we believe a hybrid force-position controller can result in more accurate probing and hence better stiffness estimation. Finally, we plan to perform experiments on ex-vivo organs and carry user studies to asses the efficacy of the system in a realistic surgical setting.

10.7 Contribution

This is a capstone chapter describing how the various contributions of the previous chapters can be combined into a single system.

10.8 Published work

Material from this chapter has appeared in the following publications

1. Nicolas Zevallos, R. Arun Srivatsan, Hadi Salman, Lu Li, Jianing Qian, Saumya Saxena, Mengyun Xu, Kartik Patath and Howie Choset, "A Real-time Aug-

mented Reality Surgical System for Overlaying Stiffness Information", conditionally accepted to the Robotics: Science and Systems (RSS), June 2018.

2. Nicolas Zevallos, R. Arun Srivatsan, Hadi Salman, Lu Li, Jianing Qian, Saumya Saxena, Mengyun Xu, Kartik Patath and Howie Choset, "A surgical system for automatic registration, stiffness mapping and dynamic image overlay", in proceedings of International Symposium on Medical Robotics (ISMR), Altanta, 2018. (Best paper finalist)

3. Lu Li, Bocheng Yu, Chen Yang, Prasad Vagdargi, R. Arun Srivatsan, and Howie Choset, "Development of an Inexpensive Tri-axial Force Sensor for Minimally Invasive Surgery", In proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2428, pg 906-913, 2017, Vancouver, BC, Canada.

4. Hadi Salman, Elif Ayvali, R. Arun Srivatsan, Yifei Ma, Nicolas Zevallos, Rashid Yasin, Long Wang, Nabil Siman and Howie Choset, "Trajectory-optimized sensing for active search of tissue abnormalities in robotic surgery", In the proceedings of the International Conference on Robotics and Automation (ICRA), Brisbane, Australia, May 2018.

Chapter 11

Discussions and future work

11.1 Summary of Contributions

This thesis developed a probabilistic framework for pose estimation which considers the problem in its most general form. We use dual quaternions and a mini-batch of measurements selected from the measurements available thus far, to develop a linear update model. We use results from stochastic filtering theory to derive the uncertainty in the linear model. Further, we develop a Bayesian filter for pose estimation that uses a combination of Bingham and Gaussian distributions to model the uncertainty in the rotation and translation parameters respectively. Our approach can use position, surface-normal and pose measurements. When the correspondence between the measurements and the model is known *a priori*, our approach guarantees global optimality. When the correspondence is unknown, our approach is only locally optimal. However, we develop a probabilistic approach that uses the state-uncertainty and the innovation of the filter to help escape local minima and increase the basin of convergence. Further, we also demonstrate a probabilistic approach for registration as few as twenty sparse point measurements. Finally, we presented a framework that uses probabilistic reasoning to estimate optimal registration using point measurements. This framework, unifies most of the existing literature on point-set registration and further provides suggestions to improve some of the existing methods.

11.2 Directions for Future Work

11.2.1 Globally optimal registration considering uncertainties

In Chapter 9, we presented an objective function in terms of pose parameters and uncertainties in pose and measurements (See Eq. 9.8). We also show that our objective function generalizes the objective function used by other global registration methods such as Go-ICP [18], and the works of Izatt *et. al.* [20], Horowitz *et. al.* [68] and Maron *et. al.* [69].However, it is worth noting that there is no prior work on globally optimal registration, to the best of our knowledge, that considers uncertainties in the prior pose parameters or sensor measurements. Since we have evidence from Chapter 6 and Chapter 9 that introducing uncertainties changes the objective function and shifts the global minimum to a different set of pose parameters; it is critical to consider them in the objective function. Therefor two future directions to pursue could be

1. Consider the objective function in Eq. 9.8, and reformulate it in a manner that we can use a branch-and-bound method similar to Go-ICP or perform convex relaxation in a manner that will allow us to use mixed integer optimization similar to Izatt *et. al.* [20].

2. Relax a key assumption that prior works on global registration make, *i.e.* each sensor point is matched to only one point in the model. We already have evidence from Chapter 9 that probabilistic matching provides improved estimates.

While we do realize that the resultant formulation would still be time consuming (as are all the other global registration approaches), we expect our formulation to produce more accurate results due to the consideration of uncertainties. Furthermore, in some applications such as photogrammetry [268] and offline 3D scanning [269], higher accuracy may be a requirement over real-time computation capabilities.

11.2.2 Dynamic pose estimation

This thesis focused on estimation of static poses. While Chapter 6 explored pose tracking by repeatedly applying static pose estimation, no process model was used. It is clear that if one were to use a process model that described the motion of the object, then the resultant filter may no longer be linear (unless the process model is exactly linear in the state). To deal with a nonlinear process model, one may resort to using an ensemble filter [270] or an unscented filter in the space of dual quaternions. It will be interesting to observe if (a) we can make approximations such as linearization or deterministic sampling to make state updates in the space of dual quaternions, and (b) compare if the approximations introduced still produce better results than using an IEKF or UKF with Euler angles and Cartesian coordinates.

11.2.3 Estimating multiple constrained poses

In applications such as human pose estimation [271], multi-object manipulation [272], reconstructing broken objects [273], etc. we need to find the pose of multiple objects that are constrained with respect to each other. While a naive approach to do so would be to run multiple pose estimators independently, a more efficient approach would be to use the estimate pose of one object to constrain the pose of the others. While there have been some deterministic approaches to incorporate the relative pose constraints in literature such as reassembly of broken objects by Huan *et. al.* [273], globally optimal point registration method of Izatt *et. al.* [20], and construction of multi-component shape models by Chintalapani *et. al.* [274, 275], we are not aware of any probabilistic approach that would consider sensor noise. One approach could be to segment the sensed point clouds and set up a mixed integer optimization with the integer variable assigning each segmented point cloud to an object model. Another approach could be a hierarchical approach that first registers the biggest object first and then removes the measurements corresponding to that object from the point cloud and continues with registering the next object.

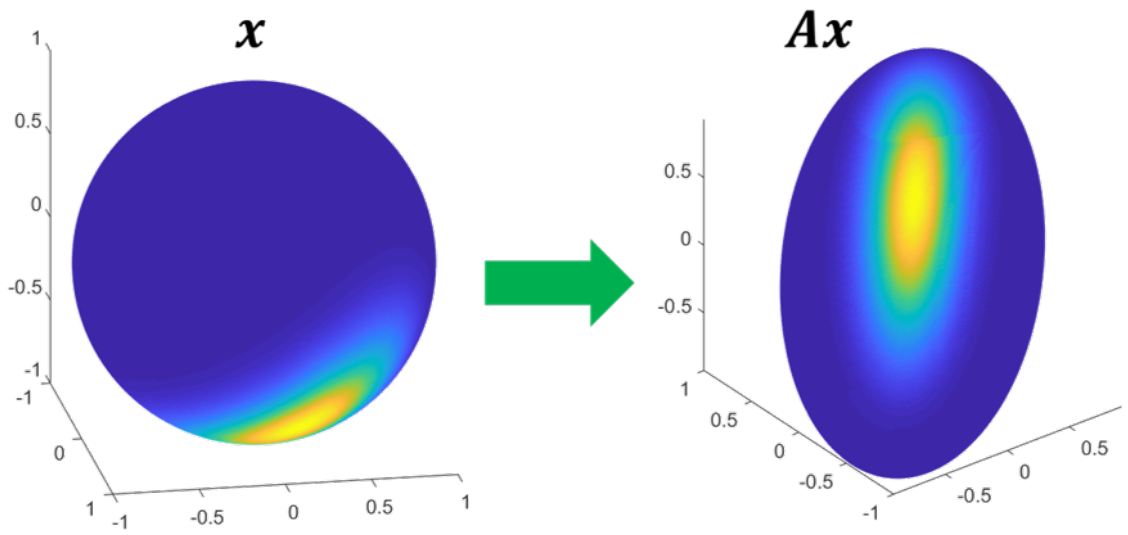


Figure 11-1: Left shows a Kent distribution from which $\boldsymbol{x}$ is sampled from. Right shows the distribution of $\boldsymbol{Ax}$.

11.2.4 Improving the choice of probability density functions when using surface-normal measurements

When dealing with surface normal measurements, a Kent distribution is a good way to describe the uncertainty. The Kent distribution is unimodal and is defined on a unit-sphere, satisfying the unit norm constraint of a surface-normal. The Kent distribution has been used to model uncertainty in surface normals by Billings *et. al.* [25, 185, 276, 221]. In this thesis we use Kent distribution to model the uncertainty in the surface normal in Chapter 6, but use an approximate upper bounded Gaussian distribution for estimating the uncertainty in the linear update model. We do this because, if a variable $\boldsymbol{x}$ has a Kent distributed uncertainty, then a linear mapping $\boldsymbol{Ax}$ does not have a Kent distributed uncertainty (see Fig. 11-1), unlike the case where $\boldsymbol{x}$ has Gaussian distributed uncertainty. We however, do observe that the uncertainty in $\boldsymbol{Ax}$ is described by a unimodal on an ellipsoid. One way to avoid using an approximate Gaussian distribution and continue using a Kent distribution is by modeling the uncertainty in the linear update model using a distribution such as the one described by Parry *et. al.* [277]. Another approach could be to use an unscented filtering over the unit-sphere similar to the approach of Gilitschenski *et. al.* [54].

11.3 Frequently Asked Questions

1. **We have ICP. Why do we need any other registration method at all?**
   ICP is a very easy to implement method for registration and it works surprisingly well in most applications. Therefore it makes sense to choose ICP as the first registration method to try in any new application. However, as pointed out by this thesis, ICP is merely a locally optimal approach that too when the sensor noise is isotropic. When the noise in the model or the measurements are anisotropic or when there exists some prior uncertainty in the registration parameters, ICP will not produce accurate results as the assumptions that ICP makes do not accurately depict the scenario. A better approach would be to use an objective function that considers the anisotropy and also uses all the available prior information to make an accurate estimation of the pose. Another related scenario where ICP would perform poorly is described in the answer to FAQ 12. Furthermore, if we were to implement the probabilistic framework developed in this thesis, then depending up on the scenario, we would automatically end up with the most optimal objective function to be minimized. This includes the ICP when the prior is unknown and noise is isotropic.

2. **What is the difference between registration and pose estimation?**
   Pose estimation is a general problem of finding an element of $SE(3)$. Registration is a specific pose estimation problem in which one needs to find the pose between a set of points (or points and surface normals) and a model (which may be composed of points and normals). Typically in registration problems one does not have the knowledge of correspondence between the measurements and the model.

3. **Why not simply use deep learning?**
   In the recent years, deep learning has gained widespread popularity in pose estimation applications [41, 37, 42, 43, 44, 45, 46, 36, 37]. However, one needs to be cautious when using deep learning methods as they typically need a large repository of labeled datasets, which may be easily available when it comes to

applications such as manipulating IKEA furniture [37] or finding pose of humans for gaming [271], etc. but not so easy when it comes to medical applications. Deep learning approaches do not provide bounds on the estimated pose and they often use an ICP in the last step, initialized with the output of the learning method. We advocate for using our probabilistic approach in the last step of the learning method instead of ICP. Further, in medical applications where accuracy and boundedness are critical, a model-based approach such as ours would provide more confidence to the user as opposed to a black-box optimization such as a deep learning approach.

4. **When to choose a probabilistic approach over a deterministic approach? Or should we always choose a probabilistic approach?**
   Deterministic approaches can be thought of as a special case of probabilistic approaches. When the uncertainties in the probabilistic approach diminishes to zero, we converge onto a deterministic approach. However, if there exist noise in the sensor measurements, but we do not know the magnitude of that noise, then using a probabilistic approach with a wrong noise uncertainty can produce results that are worse compared to a deterministic approach with no noise. The same is true for a wrong prior as well. A bad prior can be worse than using no prior at all. Thus, it is recommended that deterministic methods be used when a good estimate of the noise uncertainties is not present. If a good estimate of the noise uncertainties is present but nor prior is present then MLE methods are recommended over MAP (see Chapter 9 for more discussions).

5. **What are some real-world scenarios, where heuristic information can be used in addition to sensed point and normals for pose estimation?**
   It is an important observation that in many real-world scenarios, heuristic information is indeed present and it is important to use that information in an appropriate manner to improve the pose estimation. Here are some examples where heuristic information can help improve the pose estimation – (i) When using stereo-vision for 3D reconstruction, we obtain color information in ad-

dition to point cloud. If we use the color intensity in addition to the point location for correspondence, we can easily eliminate a large number of incorrect correspondences and improve the estimation [278]. (ii) For manipulating objects on a table top [279], we can first use RANSAC to fit a plane and estimate the surface normal direction ($\boldsymbol{z}$) to the table top and then set the prior pose uncertainties to be small in translation along $\boldsymbol{z}$ and rotation along $\boldsymbol{x}$ and $\boldsymbol{y}$. (iii) When using a Kinect for structure from motion indoors, we can use heuristic knowledge that there would exist plane walls and floors and detect them using RANSAC and use their corners and edges for correspondence [280]. (iii) When performing contact-based localization to a surgical organ, one can use anatomical segments to restrict the possible point correspondences and improve the pose estimation [70]. (iv) For laser scanning applications, we have heuristic knowledge that there are no obstacles present between the scanner and the object and thus the point cloud obtained would have very few outliers and would be a connected set. This allows us to use scale-invariant feature matching such as [27].

6. **Why does this thesis not consider heuristic information in pose estimation?**

   The focus of this thesis was to demonstrate ideas in generic setting with no help from heuristic knowledge. Not having heuristic information in a more challenging scenario and demonstrating the ideas to work in such a scenario only helps convince ourselves that the results would improve when using heuristic information. The major influence would be in the correspondence step and in setting up of the prior uncertainties.

7. **Some of the formulations appear difficult to implement or potentially time consuming. Is it worth the time and effort to understand, and continue further development on these lines, instead of using simpler and faster implementations which might compromise a little on the accuracy?**

This thesis aims to advance our understanding of pose estimation, provide a framework to unify existing methods and help decide the best approach for a given problem. This indeed comes at the cost of increased complexity of formulations. However, the complexity is complemented by improved accuracy in the estimates. Further, the formulations provide probabilistic estimates, which means in addition to finding the best pose, we also provide uncertainty in the pose which can further be used to guide collection of future measurements and provide bounds to help decide when to stop collecting measurements. If the application does not need highly accurate pose estimates or uncertainty information, it may still be meaningful to use our approach due to its wider basin of convergence.

8. **If pose estimation is ultimately an optimization problem, then why do we not use a global optimization method and call it a day? What is the advantage of putting efforts into reformulating the problem and reposing the problem in different ways?**

   Pose estimation is indeed a nonconvex optimization problem and using a global optimizer should help find the best pose. However, there are two aspects which are important to note– (i) The correct objective function should be used for optimization. For example, the one used by Go-ICP [18] does not explicitly reason about correspondence, the one used by Izatt *et. al.* [20] improves upon Go-ICP but does not consider sensor noise, the one used by Gelfand *et. al.* [27] uses curvature information which may not be available in all cases. (ii) Global optimizers are often computationally expensive and pose difficulty to use in real-time applications [18, 20].

9. **There exist prior works such as Faugeras *et. al.* [60] and Walker *et. al.* [61] which use linear models for finding the pose. How are the linear models in this work different?**

   Faugeras *et. al.* [60] decouple orientation from translation and develop a linear model in terms of unit quaternion. Walker *et. al.* [61] use dual quaternions to

parameterize the pose and derive a linear equation in terms of rotation quaternion. They then substitute the value of the calculated rotation quaternion to obtain translation. While both these approaches appear very similar to our approach, the key differences are – (i) These models do not consider sensor uncertainty or prior pose uncertainty. Incorporating the uncertainties while retaining the linear structure of the model in an important contribution of this thesis. This was possible because of modeling uncertainty in unit quaternions using a Bingham distribution and using results from stochastic filtering theory (see Prop 1). (ii) Both [60] and [61] are batch processing methods. Which means, when dealing with online pose estimation applications, these methods can be repeatedly applied on a batch of measurements obtained thus far. However, we observe in Chapters 6, 9 that such an approach does not produce good results. On the other hand, our approach updates pose and uncertainties, and produces accurate estimates because the information from the previous measurements are 'baked' into the state uncertainty. Finally, our approach becomes identical to both [60, 61] when the model uncertainty is zero, there is no prior uncertainty and the sensor uncertainty is isotropic.

10. **How are the ideas presented in this thesis different from prior probabilistic approaches such as IMLP of Billings *et. al.* [23] and filtering approaches [5, 6, 64]?**

    A key difference between IMLP and our approach is that our approach is a maximum a posteriori estimate (MAP) while IMLP is maximum likelihood estimate (MLE). IMLP does not explicitly consider prior pose uncertainty in its formulation, which our approach does. In order to account for errors in correspondence due to wrong pose estimates, IMLP incorporates a term called "match uncertainty" in the noise uncertainties. This is a dynamic term that adapts to the level of misalignment in the shapes being registered, and hence in a sense is similar to an isotropic pose uncertainty. Chapter 9 shows the benefit of considering prior pose distribution. Furthermore, similar to the answer to

the previous question, IMLP is a batch processing method and repeatedly using IMLP for online estimation produces poor results. Furthermore the methods developed in Chapter 7, 8 help deal with escaping local minima solutions and using a small number of sparse point measurements for pose estimation. These approaches require state uncertainty estimate which is not present in IMLP. The filtering-based approaches such as [5, 6, 64] are MAP estimates similar to ours, however, their update models are nonlinear and require linearization or higher order approximations. These approximations introduce errors in the estimate and can even lead to divergence due to initialization errors.

11. **What are the benefits of the methods shown in Part III, compared to using a global optimizer to overcome local minima?**

    In Part III we use the state uncertainties to develop approaches that can avoid local minima and explore the space for global optimum. This approach can be thought of as a global optimization in itself. The key difference between our approach compared to a conventional global optimizer such as genetic algorithm, particle swarm optimization, etc., is that our approach explicitly uses state uncertainty to guide the exploration and exploitation, while conventional methods do not. Using uncertainty information to guide the search for optima allows results in fewer parameters to tune and faster convergence as shown by the results in Chapter 7.

12. **Lets consider an example where there are two sources of measurements. One is accurate and the other is very noisy. Is there any benefit to using the noisy measurements at all or should we estimate the pose only using the accurate measurements?**

    In general if using an accurate sensor provides sufficient information for the pose estimation, then we can ignore the noisy sensor all together, as we would need lesser processing and computation resources. However, redundancy does have its benefits. In some cases, the noisy sensor may provide complementary information, for example, a noisy force sensor can provide surface-normal infor-

mation which could complement an accurate position sensor. Or a noisy camera looking at the object from a point of view different from that of an accurate camera can provide information that would be hidden from the view of the second camera. This information even though noisy can greatly improve the pose estimate. Even if both the sensors are providing similar information, using information from both can benefit the system in case the accurate sensor malfunctions. Setting the sensor uncertainties appropriately will ensure that while updating the pose, the contribution of the accurate sensor is weighed higher than the contribution of the inaccurate sensor as dictated by the ratio of the uncertainties. However, if one were to use registration approaches that do not consider sensor noise (such as ICP), then the results can be affected greatly by using measurements from the noisy sensor, as opposed to not using them at all.

.1 Appendix 1

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

$$\boldsymbol{\Sigma^h} = \boldsymbol{G}(\boldsymbol{\mu_x})\boldsymbol{\Sigma^b}\boldsymbol{G}^T(\boldsymbol{\mu_x}) + \boldsymbol{N}(\boldsymbol{\Sigma^b} \circledast \boldsymbol{\Sigma^x})\boldsymbol{N}^T + \boldsymbol{\Sigma^c}, \tag{1}$$

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

$$\boldsymbol{N} \triangleq [\boldsymbol{G}_1 \ \boldsymbol{G}_2 \ \cdots \ \boldsymbol{G}_m]. \tag{2}$$

$\boldsymbol{G}_i \in \mathbb{R}^{n \times m}$ is obtained from the following identity,

$$\boldsymbol{G}_i\boldsymbol{x} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{e}_i,$$

where $\boldsymbol{e}_i$ is the unit vector in $\mathbb{R}^m$ with 1 at position i and 0 everywhere else. We now use Proposition 1, to find the covariance of $\boldsymbol{h}$ which is defined as $\boldsymbol{h} = \sum_{i=1}^{p} \boldsymbol{G}_i(\boldsymbol{x})\boldsymbol{b}_i$

$$\begin{aligned} &\boldsymbol{h} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{b}, \quad \text{where,} \\ &\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1(\boldsymbol{x}) & & \\ & \ddots & \\ & & \boldsymbol{G}_p(\boldsymbol{x}) \end{bmatrix}, \quad \boldsymbol{b} = (\boldsymbol{b}_1^T, \ldots, \boldsymbol{b}_p^T)^T. \end{aligned}$$

Using Proposition 1 and simplifying, we obtain

$$\boldsymbol{\Sigma}_h = \sum_{i=1}^{p} \left[\boldsymbol{G}_i(\boldsymbol{x})\boldsymbol{\Sigma}_{\boldsymbol{b}_i}\boldsymbol{G}_i(\boldsymbol{x})^T + \boldsymbol{N}_i(\boldsymbol{\Sigma}_{\boldsymbol{b}_i} \circledast \boldsymbol{\Sigma}_{\boldsymbol{x}})\boldsymbol{N}_i^T \right], \tag{3}$$

where $\boldsymbol{N}_i$ is obtained from Eq. 2.

Example .1.1. Let us apply Proposition 1 to find uncertainty associated with $\boldsymbol{h} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{b} + \boldsymbol{c}$, where $\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2$. Let us assume,

$$\boldsymbol{\Sigma^b} = \begin{bmatrix} 0.7 & 0.01 \\ 0.01 & 4 \end{bmatrix}, \quad \boldsymbol{\Sigma^c} = \begin{bmatrix} 0.7 & 0.1 \\ 0.1 & 0.2 \end{bmatrix}, \quad \boldsymbol{G}(\boldsymbol{x}) = \begin{bmatrix} x_1 & -x_2 \\ x_2 & x_1 \end{bmatrix},$$

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

From Proposition 1, we obtain

$$\boldsymbol{\Sigma^h} = \begin{bmatrix} 1.4723 & 0.1100 \\ 0.1100 & 4.2773 \end{bmatrix}. \tag{4}$$

In order to verify the accuracy of this covariance, we perform a Monte Carlo experiment. We generate 100,000 samples of $\boldsymbol{x}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ from their respective distributions. $\boldsymbol{h}$ is evaluated for each of the samples. We then find the covariance of the resulting samples of $\boldsymbol{h}$. The calculated covariance is

$$\boldsymbol{\Sigma^h_{MC}} = \begin{bmatrix} 1.4741 & 0.1081 \\ 0.1081 & 4.2757 \end{bmatrix}. \tag{5}$$

Note that $\boldsymbol{\Sigma^h_{MC}}$ is similar to $\boldsymbol{\Sigma^h}$.

Example .1.2. Let us apply Proposition 1 to find uncertainty associated with $\boldsymbol{h} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{b} + \boldsymbol{c}$, where $\boldsymbol{x} = (x_1, x_2)^T \in \mathbb{R}^2$, and $|\boldsymbol{x}|$=1. Let us assume, the same values for all parameters, except $\boldsymbol{\Sigma^x}$. Since $\boldsymbol{x}$ is constrained to have a unit norm, we obtain $\boldsymbol{x}$ from a Bingham distribution, $\mathcal{B}(\boldsymbol{M}, \boldsymbol{Z})$ (instead of a Gaussian as in the previous example). Since the Bingham distribution is a Gaussian distribution with a unit norm constraint as shown in Sec. 2.5 and Fig. 2-3, we demonstrate with an example that Proposition 1 is valid even when $\boldsymbol{x}$ is obtained from a Bingham distribution instead of a Gaussian.

Let $\boldsymbol{M} = \boldsymbol{I}_{2\times 2}$ and $\boldsymbol{Z} = diag(0, -30)$. We first perform a Monte Carlo experiment with 100,000 of $\boldsymbol{x}, \boldsymbol{b}$ and $\boldsymbol{c}$ obtained from their respective distributions. $\boldsymbol{h}$ is evaluated

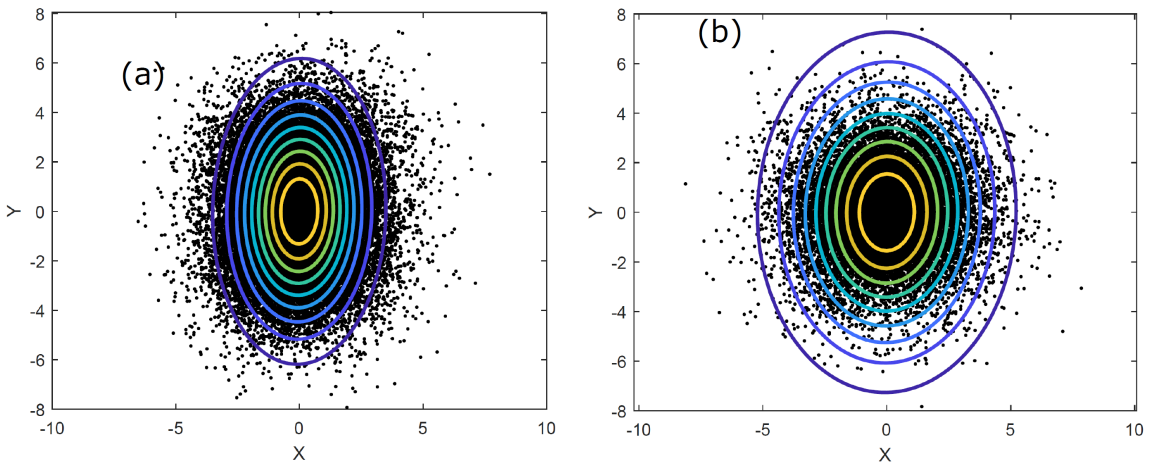


Figure -2: PDF contours obtained from the covariance estimated using Proposition 1. The black points are samples of $\boldsymbol{h} = \boldsymbol{G}(\boldsymbol{x})\boldsymbol{b} + \boldsymbol{c}$ obtained from 100,000 Monte Carlo simulations. (a) $\boldsymbol{Z} = diag(0, -30)$ (b) $\boldsymbol{Z} = diag(0, -3)$.

at each of these samples and are plotted as shown in Fig. -2. Notice that the points are distributed in the form of a Gaussian. The mean of the points is $(0.005, 0.016)^T$ and covariance is

$$\boldsymbol{\Sigma}^{\boldsymbol{h}}_{MC} = \begin{bmatrix} 1.4551 & 0.1263 \\ 0.1263 & 4.1624 \end{bmatrix}. \tag{6}$$

From Eq. 2.25, $\boldsymbol{\Sigma}^{\boldsymbol{x}} = -0.5\left(\boldsymbol{M}\left(\boldsymbol{Z} + \lambda \boldsymbol{I}\right)\boldsymbol{M}^T\right)^{-1}$. Depending on the value of λ chosen, the value of $\boldsymbol{\Sigma}^{\boldsymbol{x}}$ changes. We observe that choosing $\lambda = \min(z_i)$ results in a conservative estimate for the covariance. The estimated covariance is

$$\boldsymbol{\Sigma}^{\boldsymbol{h}} = \begin{bmatrix} 1.4450 & 0.1101 \\ 0.1101 & 4.2725 \end{bmatrix}. \tag{7}$$

$\boldsymbol{\Sigma}^{\boldsymbol{h}}_{MC}$ and $\boldsymbol{\Sigma}^{\boldsymbol{h}}$ are in good agreement with each other. Fig. -2(a) shows PDF contours for the estimated $\boldsymbol{\Sigma}^{\boldsymbol{h}}$ and the samples from the Monte Carlo simulation. We repeat this experiment for $\boldsymbol{Z} = diag(0, -3)$. The PDF contours as well as 100,000 points obtained from Monte Carlo simulation as shown in Fig. -2(b). Note how the estimated PDF accurately describes distribution of the point samples obtained from Monte Carlo simulation.

.2 Appendix 2

Estimating the likelihood of obtaining $\boldsymbol{t}$

From Eq. 6.7,

$$\widetilde{\boldsymbol{t}} = \frac{(\widetilde{\boldsymbol{a}}_1 + \widetilde{\boldsymbol{a}}_2)}{2} - \widetilde{\boldsymbol{q}} \odot \frac{(\widetilde{\boldsymbol{b}}_1 - \widetilde{\boldsymbol{b}}_2)}{2} \odot \widetilde{\boldsymbol{q}}^*, \tag{8}$$

$$= \widetilde{\boldsymbol{a}}_c - \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_c \odot \widetilde{\boldsymbol{q}}^*, \tag{9}$$

$$\Rightarrow \widetilde{\boldsymbol{t}} \odot \widetilde{\boldsymbol{q}} = \widetilde{\boldsymbol{a}}_c \odot \widetilde{\boldsymbol{q}} - \widetilde{\boldsymbol{q}} \odot \widetilde{\boldsymbol{b}}_c,$$

$$\Rightarrow \boldsymbol{F}_2(\widetilde{\boldsymbol{q}})\widetilde{\boldsymbol{t}} = \boldsymbol{F}_2(\widetilde{\boldsymbol{q}})\widetilde{\boldsymbol{a}}_c - \boldsymbol{F}_1(\widetilde{\boldsymbol{q}})\widetilde{\boldsymbol{b}}_c, \quad \text{from Eq. 2.11}$$

$$\Rightarrow \boldsymbol{W}_1\boldsymbol{t} = \boldsymbol{W}_1\boldsymbol{a}_c - \boldsymbol{W}_2\boldsymbol{b}_c, \tag{10}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{4\times3}$ are the last three columns of $\boldsymbol{F}_2(\widetilde{\boldsymbol{q}})$ and $\boldsymbol{F}_1(\widetilde{\boldsymbol{q}})$ respectively.

From Sec. 6.2.1, $\boldsymbol{a}_i^s = \boldsymbol{a}_i + \delta\boldsymbol{a}_i$ and $\boldsymbol{b}_i^s = \boldsymbol{b}_i + \delta\boldsymbol{b}_i$. Substituting these terms in Eq. 10 gives

$$\boldsymbol{W}_1\boldsymbol{t} - \boldsymbol{W}_1\boldsymbol{a}_c^s + \boldsymbol{W}_2\boldsymbol{b}_c^s + \boldsymbol{G}_2(\widetilde{\boldsymbol{q}})\boldsymbol{\mu} = 0,$$

where $\boldsymbol{\mu} = (\delta\boldsymbol{a}_1, \delta\boldsymbol{a}_2, \delta\boldsymbol{b}_1, \delta\boldsymbol{b}_2)^T$ and

$$\boldsymbol{G}_2 = \frac{1}{2}\begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_1 & -\boldsymbol{W}_2 & -\boldsymbol{W}_2\end{bmatrix}$$

As observed earlier in the case of estimation of $\widetilde{\boldsymbol{q}}$, $\boldsymbol{G}_2(\widetilde{\boldsymbol{q}})\boldsymbol{\mu}$ is a zero mean Gaussian noise, $\mathcal{N}(0, \boldsymbol{R})$, where the uncertainty $\boldsymbol{R}$ is obtained analytically similar to the evaluation of $\boldsymbol{Q}$ from Proposition 1.

Bibliography

[1] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, "The banana distribution is gaussian: A localization study with exponential coordinates," *Robotics: Science and Systems VIII; MIT Press: Cambridge, MA, USA*, pp. 265–272, 2013.

[2] E. Ayvali, R. A. Srivatsan, L. Wang, R. Roy, N. Simaan, and H. Choset, "Using Bayesian Optimization to Guide Probing of a Flexible Environment for Simultaneous Registration and Stiffness Mapping," in *ICRA*, pp. 931–936, 2016.

[3] S. Tully, G. Kantor, M. Zenati, and H. Choset, "Shape estimation for image-guided surgery with a highly articulated snake robot," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 1353–1358, 2011.

[4] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, no. 4, pp. 629–642, 1987.

[5] 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.

[6] 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.

[7] G. Turk and M. Levoy, "The Stanford 3D Scanning Repository," *Stanford University Computer Graphics Laboratory http://graphics.stanford.edu/data/3Dscanrep*, 2005.

[8] L. Li, B. Yu, C. Yang, P. Vagdargi, R. A. Srivatsan, and H. Choset, "Development of an Inexpensive Tri-axial Force Sensor for Minimally Invasive Surgery," in *In proceedings of the International Conference on Intelligent Robots and Systems*, IEEE, 2017.

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

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

[11] P. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, Feb 1992.

[12] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, IEEE, 2001.

[13] J. Glover, G. Bradski, and R. B. Rusu, "Monte carlo pose estimation with quaternion kernels and the distribution," in *Robotics: Science and Systems*, vol. 7, p. 97, 2012.

[14] B. D. Lucas, T. Kanade, *et al.*, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, vol. 81, pp. 674–679, 1981.

[15] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, *et al.*, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

[16] B. Ma and R. Ellis, "Surface-based registration with a particle filter," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 566–573, 2004.

[17] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 698–700, 1987.

[18] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3d registration efficiently and globally optimally," in *2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 1457–1464, Dec 2013.

[19] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast Global Registration," in *European Conference on Computer Vision*, pp. 766–782, Springer, 2016.

[20] G. Izatt, H. Dai, and R. Tedrake, "Globally Optimal Object Pose Estimation in Point Clouds with Mixed-Integer Programming," in *International Symposium on Robotics Research*, 12 2017.

[21] R. S. J. Estépar, A. Brun, and C.-F. Westin, "Robust generalized total least squares iterative closest point registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2004.

[22] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: science and systems*, vol. 2, p. 435, 2009.

[23] 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, p. e0117688, 2015.

[24] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 569–585, 2011.

[25] S. Billings and R. Taylor, "Iterative most likely oriented point registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 178–185, Springer, 2014.

[26] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3, pp. 282–285, IEEE, 2001.

[27] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on geometry processing*, vol. 2, p. 5, 2005.

[28] K. Pulli, "Multiview registration for large data sets," in *Proceedings. Second International Conference on 3-D Digital Imaging and Modeling, 1999.*, pp. 160–168, IEEE, 1999.

[29] "A modified ICP algorithm for normal-guided surface registration,"

[30] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 345–358, 1989.

[31] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research (IJRR)*, vol. 14, no. 3, pp. 195–210, 1995.

[32] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[33] M. K. Ackerman, A. Cheng, B. Shiffman, E. Boctor, and G. Chirikjian, "Sensor calibration with unknown correspondence: Solving AX= XB using Euclidean-group invariants," in *International Conference on Intelligent Robots and Systems (IROS)*, pp. 1308–1313, IEEE, 2013.

[34] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-time articulated hand pose estimation using semi-supervised transductive regression forests," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 3224–3231, IEEE, 2013.

[35] J. Vongkulbhisal, F. De la Torre, and J. P. Costeira, "Discriminative optimization: theory and applications to point cloud registration," in *IEEE CVPR*, 2017.

[36] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, and T.-K. Kim, "Pose Guided RGBD Feature Learning for 3D Object Pose Estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3856–3864, 2017.

[37] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2686–2694, 2015.

[38] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A convolutional network for real-time 6-DOF camera relocalization," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, IEEE, 2015.

[39] F. Massa, R. Marlet, and M. Aubry, "Crafting a multi-task CNN for viewpoint estimation," *arXiv preprint arXiv:1609.03894*, 2016.

[40] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese, "Objectnet3D: A large scale database for 3d object recognition," in *European Conference on Computer Vision*, pp. 160–176, Springer, 2016.

[41] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3D bounding box estimation using deep learning and geometry," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5632–5640, IEEE, 2017.

[42] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," *arXiv preprint arXiv:1711.08848*, 2017.

[43] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1529, 2017.

[44] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[45] M. Rad and V. Lepetit, "BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth," in *International Conference on Computer Vision*, 2017.

[46] C. Li, J. Bai, and G. D. Hager, "A Unified Framework for Multi-View Multi-Class Object Pose Estimation," *arXiv preprint arXiv:1803.08103*, 2018.

[47] S. Tully, G. Kantor, and H. Choset, "Inequality constrained Kalman filtering for the localization and registration of a surgical robot," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5147–5152, IEEE, 2011.

[48] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[49] 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.

[50] 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.

[51] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[52] L. Carlone and A. Censi, "From angular manifolds to the integer lattice: Guaranteed orientation estimation with application to pose graph optimization," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 475–492, 2014.

[53] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group," *arXiv preprint arXiv:1612.07386*, 2016.

[54] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "Unscented orientation estimation based on the Bingham distribution," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 172–177, 2016.

[55] I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck, "A new probability distribution for simultaneous representation of uncertain position and orientation," in *17th International Conference on Information Fusion*, pp. 1–7, IEEE, 2014.

[56] T. Fan, H. Weng, and T. Murphey, "Decentralized and Recursive Identification for Cooperative Manipulation of Unknown Rigid Body with Local Measurements," *arXiv preprint arXiv:1709.01555*, 2017.

[57] C. Bingham, "An antipodally symmetric distribution on the sphere," *The Annals of Statistics*, pp. 1201–1225, 1974.

[58] J.-H. Kim and R. Kumar, Vijay, "Kinematics of Robot Manipulators via Line Transformations," *Journal of Field Robotics*, vol. 7, no. 4, pp. 649–674, 1990.

[59] M. Shoham and F.-H. Jen, "On rotations and translations with application to robot manipulators," *Advanced Robotics*, vol. 8, no. 2, pp. 203–229, 1993.

[60] O. D. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D objects," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 27–52, 1986.

[61] M. W. Walker, L. Shao, and R. A. Volz, "Estimating 3-D location parameters using dual number quaternions," *CVGIP: image understanding*, vol. 54, no. 3, pp. 358–367, 1991.

[62] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda, "An extended Kalman filter for quaternion-based orientation estimation using MARG sensors," in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 2003–2011, IEEE, 2001.

[63] J. J. LaViola Jr, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proceedings of the American Control Conference*, vol. 3, pp. 2435–2440, IEEE, 2003.

[64] J. S. Goddard and M. A. Abidi, "Pose and motion estimation using dual quaternion-based extended Kalman filtering," in *Three-Dimensional Image Capture and Applications*, vol. 3313, pp. 189–201, International Society for Optics and Photonics, 1998.

[65] M. Brossard, S. Bonnabel, and A. Barrau, "Unscented Kalman Filter on Lie Groups for Visual Inertial Odometry," 2018.

[66] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[67] D. M. Rosen, L. Carlone, A. S. Bandeira, and J. J. Leonard, "A certifiably correct algorithm for synchronization over the special Euclidean group," *12th International Workshop on Agorithmic Foundations of Robotics*, 2016.

[68] M. B. Horowitz, N. Matni, and J. W. Burdick, "Convex relaxations of SE(2) and SE(3) for visual pose estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1148–1154, IEEE, 2014.

[69] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman, "Point registration via efficient convex relaxation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 73, 2016.

[70] 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.

[71] B. Ma and R. E. Ellis, "Robust registration for computer-integrated orthopedic surgery: laboratory validation and clinical experience," *Medical image analysis*, vol. 7, no. 3, pp. 237–250, 2003.

[72] B. Ma and R. E. Ellis, "Spatial-stiffness analysis of surface-based registration," in *MICCAI*, pp. 623–630, 2004.

[73] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 1297–1304, IEEE, 2006.

[74] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *IEEE International Conference on Robotics and Automation*, pp. 3212–3217, IEEE, 2009.

[75] G. Godin, M. Rioux, and R. Baribeau, "Three-dimensional registration using range and intensity information," in *Videometrics III*, vol. 2350, pp. 279–291, International Society for Optics and Photonics, 1994.

[76] N. D. Cornea, M. F. Demirci, D. Silver, S. Dickinson, P. Kantor, *et al.*, "3D object retrieval using many-to-many matching of curve skeletons," in *Shape Modeling and Applications, 2005 International Conference*, pp. 366–371, IEEE, 2005.

[77] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, "One point isometric matching with the heat kernel," in *Computer Graphics Forum*, vol. 29, pp. 1555–1564, Wiley Online Library, 2010.

[78] S. Sanan, S. Tully, A. Bajo, N. Simaan, and H. Choset, "Simultaneous Compliance and Registration Estimation for Robotic Surgery," in *Proceedings of the Robotics: Science and Systems Conference*, 2014.

[79] S. Chen, "Kalman filter for robot vision: a survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.

[80] R. Toscano and P. Lyonnet, "Heuristic Kalman algorithm for solving optimization problems.," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 39, no. 5, pp. 1231–1244, 2009.

[81] G. Kurz, I. Gilitschenski, and U. D. Hanebeck, "Recursive nonlinear filtering for angular data based on circular distributions," in *American Control Conference (ACC), 2013*, pp. 5439–5445, IEEE, 2013.

[82] J. Luck, C. Little, and W. Hoff, "Registration of range data using a hybrid simulated annealing and iterative closest point algorithm," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3739–3744, IEEE, 2000.

[83] F. L. Seixas, L. S. Ochi, A. Conci, and D. M. Saade, "Image registration using genetic algorithms," in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 1145–1146, ACM, 2008.

[84] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[85] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *European Conference on Computer Vision*, pp. 418–432, Springer, 2002.

[86] 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," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 7, pp. 1199–1217, 2013.

[87] J. M. Selig, *Geometric fundamentals of robotics*. New York: Springer, second ed., 1996.

[88] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC Press, 1994.

[89] G. Meurant, *Introduction to Lie groups and Lie algebras*, vol. 51. Academic Press, 1973.

[90] A. L. Onishchik, E. Vinberg, and V. Minachin, *Lie groups and Lie algebras*. Springer, 1993.

[91] J.-L. Blanco, "A tutorial on se (3) transformation parameterizations and on-manifold optimization," *University of Malaga, Tech. Rep*, vol. 3, 2010.

[92] R. Gilmore, *Lie groups, Lie algebras, and some of their applications*. Courier Corporation, 2012.

[93] F. L. Markley, Y. Cheng, L. Crassidis, John, and Y. Oshman, "Averaging quaternions," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 4, 2007.

[94] Clifford, "Preliminary Sketch of Biquaternions," *Proceedings of the London Mathematical Society*, vol. s1-4, no. 1, pp. 381–395, 1873.

[95] A. T. Yang and F. Freudenstein, "Application of dual-number quaternion algebra to the analysis of spatial mechanisms," *Journal of Applied Mechanics*, vol. 31, no. 2, pp. 300–308, 1964.

[96] G. Leclercq, P. Lefevre, and G. Blohm, "3d kinematics using dual quaternions: theory and applications in neuroscience," *Frontiers in Behavioral Neuroscience*, vol. 7, pp. 1–25, 2013.

[97] K. Shoemake, "Animating rotation with quaternion curves," *Computer Graphics*, vol. 19, no. 3, pp. 245–254, 1985.

[98] L. Kavan, S. Collins, C. O'Sullivan, and J. Zara, "Dual quaternions for rigid transformation blending," *Technical report, Trinity college Dublin*, 2006.

[99] K. Daniilidis and E. Bayro-Corrochano, "The dual quaternion approach to hand-eye calibration," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, vol. 1, pp. 318–322, IEEE, 1996.

[100] Y. Wang, Y. Wang, K. Wu, H. Yang, and H. Zhang, "A dual quaternion-based, closed-form pairwise registration algorithm for point clouds," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 94, pp. 63–69, 2014.

[101] N. Filipe, M. Kontitsis, and P. Tsiotras, "Extended kalman filter for spacecraft pose estimation using dual quaternions," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 9, pp. 1625–1641, 2015.

[102] B. Kenwright, *A beginners guide to dual-quaternions: what they are, how they work, and how to use them for 3D character hierarchies*. Václav Skala-UNION Agency, 2012.

[103] K. Kunze and H. Schaeben, "The Bingham distribution of quaternions and its spherical radon transform in texture analysis," *Mathematical Geology*, vol. 36, no. 8, pp. 917–943, 2004.

[104] T. S. Haines and R. C. Wilson, "Belief propagation with directional statistics for solving the shape-from-shading problem," in *European Conference on Computer Vision*, pp. 780–791, Springer, 2008.

[105] J. Glover and L. P. Kaelbling, "Tracking 3-d rotations with the quaternion bingham filter," 2013.

[106] M. V. Wyawahare, P. M. Patil, H. K. Abhyankar, *et al.*, "Image registration techniques: An overview," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 3, pp. 11–28, 2009.

[107] M. Hayashibe, N. Suzuki, and Y. Nakamura, "Laser-scan endoscope system for intraoperative geometry acquisition and surgical robot safety management," *Medical Image Analysis*, vol. 10, no. 4, pp. 509 – 519, 2006.

[108] R. E. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, pp. 71–87, 1 2013.

[109] T. Yamamoto, M. Bernhardt, A. Peer, M. Buss, and M. Okamura, Allison, "Techniques for Environment Parameter Estimation During Telemanipulation," in *Proceedings of the International Conference on Biomedical Robotics and Biomechatronics*, pp. 217–223, 2008.

[110] H. Liu, D. P. Noonan, B. J. Challacombe, P. Dasgupta, L. D. Seneviratne, and K. Althoefer, "Rolling mechanical imaging for tissue abnormality localization during minimally invasive surgery," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 404–414, 2010.

[111] P. Chalasani, L. Wang, R. Roy, N. Simaan, R. H. Taylor, and M. Kobilarov, "Concurrent nonparametric estimation of organ geometry and tissue stiffness using continuous adaptive palpation," in *ICRA*, pp. 4164–4171, IEEE, 2016.

[112] P. Chalasani, L. Wang, R. Yasin, N. Simaan, and R. H. Taylor, "Preliminary Evaluation of an Online Estimation Method for Organ Geometry and Tissue Stiffness," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1816–1823, 2018.

[113] 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 *IEEE International Conference on Robotics and Automation*, pp. 924–930, 2016.

[114] L. Wang, Z. Chen, P. Chalasani, R. M. Yasin, P. Kazanzides, R. H. Taylor, and N. Simaan, "Force-Controlled Exploration for Updating Virtual Fixture Geometry in Model-Mediated Telemanipulation," *Journal of Mechanisms and Robotics*, vol. 9, no. 2, p. 021010, 2017.

[115] M. Feuerstein, T. Mussack, S. Heining, and N. Navab, "Intraoperative laparoscope augmentation for port placement and resection planning in minimally invasive liver resection," *IEEE Transactions on Medical Imaging*, vol. 27, pp. 355–369, March 2008.

[116] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2262–2275, Dec 2010.

[117] R. Sagawa, K. Akasaka, Y. Yagi, and L. V. Gool, "Elastic convolved ICP for the registration of deformable objects," in *IEEE 12th International Conference on Computer Vision Workshop*, pp. 1558–1565, 2009.

[118] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable Medical Image Registration: A Survey," *Medical Imaging, IEEE Transactions on*, vol. 32, pp. 1153–1190, July 2013.

[119] T. Lange, S. Eulenstein, M. Hnerbein, and P.-M. Schlag, "Vessel-Based Non-Rigid Registration of MR/CT and 3D Ultrasound for Navigation in Liver Surgery," *Computer Aided Surgery*, vol. 8, no. 5, pp. 228–240, 2003. PMID: 15529952.

[120] A. Roche, X. Pennec, G. Malandain, and N. Ayache, "Rigid Registration of 3D Ultrasound with MR Images: a New Approach Combining Intensity and Gradient Information," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 1038–1049, 2001.

[121] K. Lunn, K. Paulsen, D. Roberts, F. Kennedy, A. Hartov, and J. West, "Displacement estimation with co-registered ultrasound for image guided neurosurgery: A quantitative in vivo porcine study," *IEEE Transactions on Medical Imaging*, vol. 22, pp. 1358–1368, Nov 2003.

[122] S. Ji, D. W. Roberts, A. Hartov, and K. D. Paulsen, "Intraoperative patient registration using volumetric true 3D ultrasound without fiducials," *Medical Physics*, vol. 39, no. 12, pp. 7540–7552, 2012.

[123] A. G. Konheim, *Hashing in computer science: Fifty years of slicing and dicing*. John Wiley & Sons, 2010.

[124] X. Kang, W.-P. Yau, and R. H. Taylor, "Simultaneous pose estimation and patient-specific model reconstruction from single image using maximum penalized likelihood estimation (MPLE)," *Pattern Recognition*, vol. 57, pp. 61–69, 2016.

[125] N. Ryan, C. Heneghan, and P. de Chazal, "Registration of digital retinal images using landmark correspondence by expectation maximization," *Image and Vision Computing*, vol. 22, no. 11, pp. 883–898, 2004.

[126] S. Chatterjee and A. S. Hadi, "Influential observations, high leverage points, and outliers in linear regression," *Statistical Science*, pp. 379–393, 1986.

[127] J. Gentleman and M. Wilk, "Detecting outliers. II. Supplementing the direct analysis of residuals," *Biometrics*, pp. 387–410, 1975.

[128] R. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, no. 1, pp. 71–87, 2013.

[129] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[130] P. Chalasani, A. Deguet, P. Kazanzides, and R. H. Taylor, "A Computational Framework for Complementary Situational Awareness (CSA) in Surgical Assistant Robots," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*, pp. 9–16, IEEE, 2018.

[131] A. Deguet, R. Kumar, R. Taylor, and P. Kazanzides, "The cisst libraries for computer assisted intervention systems," in *MICCAI Workshop*, 2008.

[132] M. Y. Jung, B. Marcin, D. Anton, R. H. Taylor, and K. Peter, "Lessons learned from the development of component-based medical robot systems," *JOURNAL OF SOFTWARE ENGINEERING IN ROBOTICS*, vol. 5, no. 2, pp. 25–41, 2014.

[133] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[134] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci® Surgical System," in *ICRA*, pp. 6434–6439, IEEE, 2014.

[135] C. A. Linte, J. Moore, and T. M. Peters, "How accurate is accurate enough? A brief overview on accuracy considerations in image-guided cardiac interventions," in *Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 2313–2316, IEEE, 2010.

[136] D. Vandeweyer, E. L. Neo, J. W. Chen, G. J. Maddern, T. G. Wilson, and R. T. Padbury, "Influence of resection margin on survival in hepatic resections for colorectal liver metastases," *HPB*, vol. 11, no. 6, pp. 499–504, 2009.

[137] E. Ayvali, A. Ansari, L. Wang, N. Simaan, and H. Choset, "Utility-Guided Palpation for Locating Tissue Abnormalities," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 864–871, 2017.

[138] H. Salman, E. Ayvali, R. A. Srivatsan, Y. Ma, N. Zevallos, R. Yasin, L. Wang, N. Simaan, and H. Choset, "Trajectory-Optimized Sensing for Active Search of Tissue Abnormalities in Robotic Surgery," in *Accepted for publication in ICRA*, IEEE, 2018.

[139] Y. Horiguchi, B. Sekoguchi, H. Imai, T. Suzuki, H. Kubo, H. Itoh, and M. Itoh, "Treatment of choice for unresectable small liver cancer: Percutaneous ethanol injection therapy or transarterial chemoembolization therapy," *Cancer chemotherapy and pharmacology*, vol. 33, no. 1, pp. S111–S114, 1994.

[140] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3d data," *Image and vision computing*, vol. 17, no. 2, pp. 135–147, 1999.

[141] L. Cerman, A. Sugimoto, and I. Shimizu, *3D shape registration with estimating illumination and photometric properties of a convex object*. na, 2007.

[142] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[143] D. Choukroun, I. Bar-Itzhack, and Y. Oshman, "Novel quaternion Kalman filter," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, pp. 174–190, Jan 2006.

[144] M. D. Shuster, "The quaternion in Kalman filtering," *Advances in the Astronautical Sciences*, vol. 85, pp. 25–37, 1993.

[145] D. Spinello and D. J. Stilwell, "Nonlinear estimation with state-dependent Gaussian observation noise," *IEEE Transactions on Automatic Control*, vol. 55, no. 6, pp. 1358–1366, 2010.

[146] A. H. Jazwinski, *Stochastic processes and filtering theory*. Courier Corporation, 2007.

[147] S. Winkelbach, S. Molkenstruck, and F. M. Wahl, "Low-cost laser range scanner and fast surface registration approach," in *Pattern Recognition*, pp. 718–728, Springer, 2006.

[148] E. Mair, M. Fleps, M. Suppa, and D. Burschka, "Spatio-temporal initialization for IMU to camera registration," in *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 557–564, IEEE, 2011.

[149] F. C. Park and B. J. Martin, "Robot sensor calibration: solving AX= XB on the Euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, 1994.

[150] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 145–151, IEEE, 1991.

[151] Z. Zhang, L. Zhang, and G.-Z. Yang, "A computationally efficient method for hand–eye calibration," *International journal of computer assisted radiology and surgery*, vol. 12, no. 10, pp. 1775–1787, 2017.

[152] T. Lefebvre, H. Bruyninckx, and J. De Schutter, *Nonlinear Kalman filtering for force-controlled robot tasks*. Springer-Verlag Berlin, 2004.

[153] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.

[154] I. Bar-Itzhack, J. Deutschmann, and F. Markley, "Quaternion normalization in additive EKF for spacecraft attitude determination," *Mechanics Estimation Theory Symposium*, p. 403, 1991.

[155] K. Nicewarner and A. Sanderson, "A general representation for orientational uncertainty using random unit quaternions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1161–1168 vol.2, May 1994.

[156] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX= XB," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[157] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley, "Rotation averaging with application to camera-rig calibration," in *Proceedings of the 9th Asian conference on Computer Vision-Volume Part II*, pp. 335–346, Springer-Verlag, 2009.

[158] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[159] M. Zefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 576–589, 1998.

[160] F. C. Park, "The optimal kinematic design of mechanisms," 1991.

[161] W. Park, Y. Liu, Y. Zhou, M. Moses, and G. S. Chirikjian, "Kinematic state estimation and motion planning for stochastic nonholonomic systems using the exponential map," *Robotica*, vol. 26, no. 4, pp. 419–434, 2008.

[162] K. C. Wolfe and M. Mashner, "Bayesian fusion on lie groups," *Journal of Algebraic Statistics*, vol. 2, no. 1, 2011.

[163] Y. Wang and G. Chirikjian, "Error Propagation on the Euclidean Group With Applications to Manipulator Kinematics," *IEEE Transactions on Robotics*, vol. 22, pp. 591–602, August 2006.

[164] S. Marandi and V. Modi, "A preferred coordinate system and the associated orientation representation in attitude dynamics," *Acta Astronautica*, vol. 15, no. 11, pp. 833–843, 1987.

[165] T. D. Barfoot and P. T. Furgale, "Associating uncertainty with three-dimensional poses for use in estimation problems," *IEEE Transactions on Robotics*, vol. 30, no. 3, pp. 679–693, 2014.

[166] G. Bourmaud, R. Megret, A. Giremus, and Y. Berthoumieu, "Discrete extended kalman filter on lie groups," in *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*, pp. 1–5, IEEE, 2013.

[167] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1, pp. 351–357, IEEE, 2004.

[168] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 209–225, 2010.

[169] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[170] G. S. Chirikjian and J. W. Burdick, "A hyper-redundant manipulator," *Robotics & Automation Magazine, IEEE*, vol. 1, no. 4, pp. 22–29, 1994.

[171] G. S. Fischer, I. Iordachita, C. Csoma, J. Tokuda, S. P. DiMaio, C. M. Tempany, N. Hata, and G. Fichtinger, "Mri-compatible pneumatic robot for transperineal prostate needle placement," *IEEE/ASME transactions on mechatronics*, vol. 13, no. 3, pp. 295–305, 2008.

[172] F. Taffoni, D. Formica, P. Saccomandi, G. D. Pino, and E. Schena, "Optical fiber-based mr-compatible sensors for medical applications: An overview," *Sensors*, vol. 13, no. 10, pp. 14105–14120, 2013.

[173] S. J. Mihailov, "Fiber bragg grating sensors for harsh environments," *Sensors*, vol. 12, no. 2, pp. 1898–1918, 2012.

[174] K. Cleary, H. Zhang, N. Glossop, E. Levy, B. Wood, and F. Banovac, "Electromagnetic tracking for image-guided abdominal procedures: Overall system and technical issues," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 6748–6753, IEEE, 2005.

[175] H. Talib, M. Styner, T. Rudolph, and G. Ballester, "Dynamic registration using ultrasound for anatomical referencing," in *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pp. 1164–1167, IEEE, 2007.

[176] A. B. Koolwal, F. Barbagli, C. Carlson, and D. Liang, "An ultrasound-based localization algorithm for catheter ablation guidance in the left atrium," *The International Journal of Robotics Research*, vol. 29, no. 6, pp. 643–665, 2010.

[177] A. Bry, A. Bachrach, and N. Roy, "State estimation for aggressive flight in GPS-denied environments using onboard sensing," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, IEEE, 2012.

[178] A. Degani, H. Choset, A. Wolf, and M. A. Zenati, "Highly articulated robotic probe for minimally invasive surgery," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 4167–4172, IEEE, 2006.

[179] K. Lister, Z. Gao, and J. Desai, "Development of In Vivo Constitutive Models for Liver: Application to Surgical Simulation," *Annals of Biomedical Engineering*, vol. 39, no. 3, pp. 1060–1073, 2011.

[180] Y. Fung, *Biomechanics: Mechanical Properties of Living Tissues*. Springer, second ed., 1993.

[181] E. Wan and R. Van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pp. 153–158, 2000.

[182] K. Feng, J. Li, X. Zhang, C. Shen, Y. Bi, T. Zheng, and J. Liu, "A new quaternion-based kalman filter for real-time attitude estimation using the two-step geometrically-intuitive correction algorithm," *Sensors*, vol. 17, no. 9, p. 2146, 2017.

[183] T. Lefebvre, H. Bruyninckx, and J. De Schuller, "A new method for the nonlinear transformation of means and covariances in filters and estimators," *IEEE Transactions on Automatic Control*, vol. 47, no. 8, pp. 1406–1409, 2002.

[184] J. Steinbring and U. D. Hanebeck, "S 2 KF: The smart sampling Kalman filter," in *16th International Conference on Information Fusion (FUSION)*, pp. 2089–2096, IEEE, 2013.

[185] S. Billings and R. Taylor, "Generalized iterative most likely oriented-point (G-IMLOP) registration," *International journal of computer assisted radiology and surgery*, vol. 10, no. 8, pp. 1213–1226, 2015.

[186] "On Kalman filtering with nonlinear equality constraints,"

[187] P. Richards, "Constrained Kalman filtering using pseudo-measurements," in *IEEE Colloquium on Algorithms for Target Tracking*, pp. 75–79, 1995.

[188] T. L. Chia, P.-C. Chow, and H. J. Chizeck, "Recursive parameter identification of constrained systems: An application to electrically stimulated muscle," *IEEE Transactions on Biomedical Engineering*, vol. 38, no. 5, pp. 429–442, 1991.

[189] T. Hillen, K. J. Painter, A. C. Swan, and A. D. Murtha, "Moments of von Mises and Fisher distributions and applications," *Mathematical biosciences and engineering: MBE*, vol. 14, no. 3, pp. 673–694, 2017.

[190] N. H. Balov, "Covariance of centered distributions on manifold," *arXiv preprint arXiv:0805.0732*, 2008.

[191] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX= YB," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.

[192] A. Tabb and K. M. A. Yousef, "Parameterizations for reducing camera reprojection error for robot-world hand-eye calibration," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 3030–3037, IEEE, 2015.

[193] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software (TOMS)*, vol. 3, no. 3, pp. 209–226, 1977.

[194] M. Blum, R. W. Floyd, V. R. Pratt, R. L. Rivest, and R. E. Tarjan, "Time bounds for selection," *J. Comput. Syst. Sci.*, vol. 7, no. 4, pp. 448–461, 1973.

[195] S. M. Omohundro, *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989.

[196] P. N. Yianilos, "Data structures and algorithms for nearest neighbor search in general metric spaces," in *SODA*, vol. 93, pp. 311–321, 1993.

[197] J. Williams, R. Taylor, and L. Wolff, "Augmented kd techniques for accelerated registration and distance measurement of surfaces," *Computer Aided Surgery: Computer-Integrated Surgery of the Head and Spine*, pp. 1–21, 1997.

[198] N. Verma, S. Kpotufe, and S. Dasgupta, "Which spatial partition trees are adaptive to intrinsic dimension?," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pp. 565–574, AUAI Press, 2009.

[199] S. Dasgupta and Y. Freund, "Random projection trees and low dimensional manifolds," in *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pp. 537–546, ACM, 2008.

[200] M. McCartin-Lim, A. McGregor, and R. Wang, "Approximate principal direction trees," *arXiv preprint arXiv:1206.4668*, 2012.

[201] J. Wang, N. Wang, Y. Jia, J. Li, G. Zeng, H. Zha, and X.-S. Hua, "Trinary-projection trees for approximate nearest neighbor search," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 2, pp. 388–403, 2014.

[202] N. Paragios, M. Rousson, and V. Ramesh, "Non-rigid registration using distance functions," *Computer Vision and Image Understanding*, vol. 89, no. 2-3, pp. 142–165, 2003.

[203] S. Flöry and M. Hofer, "Surface fitting and registration of point clouds using approximations of the unsigned distance function," *Computer Aided Geometric Design*, vol. 27, no. 1, pp. 60–77, 2010.

[204] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[205] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *Proc. of the International Conference on Intelligent Robot Systems*, Oct. 2012.

[206] A. Neumaier, O. Shcherbina, W. Huyer, and T. Vinkó, "A comparison of complete global optimization solvers," *Mathematical programming*, vol. 103, no. 2, pp. 335–356, 2005.

[207] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[208] S. P. Brooks and B. J. Morgan, "Optimization using simulated annealing," *The Statistician*, pp. 241–257, 1995.

[209] J. H. Holland, "Outline for a logical theory of adaptive systems," *Journal of the ACM*, vol. 9, no. 3, pp. 297–314, 1962.

[210] R. Toscano and P. Lyonnet, "A Kalman Optimization Approach for Solving Some Industrial Electronics Problems," *IEEE Transactions on Industrial Electronics*, vol. 11, no. 59, pp. 4456–4464, 2012.

[211] F. Schoen, "Stochastic techniques for global optimization: A survey of recent advances," *Journal of Global Optimization*, vol. 1, no. 3, pp. 207–228, 1991.

[212] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm intelligence*, vol. 1, no. 1, pp. 33–57, 2007.

[213] A. L. Friesen and P. Domingos, "Recursive decomposition for nonconvex optimization," in *Proceedings of IJCAI*, 2015.

[214] L. Ingber, "Simulated annealing: Practice versus theory," *Mathematical and computer modelling*, vol. 18, no. 11, pp. 29–57, 1993.

[215] D. J. Ram, T. Sreenivas, and K. G. Subramaniam, "Parallel simulated annealing algorithms," *Journal of parallel and distributed computing*, vol. 37, no. 2, pp. 207–212, 1996.

[216] T. Schön, F. Gustafsson, and A. Hansson, "A note on state estimation as a convex optimization problem," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, pp. VI–61, IEEE, 2003.

[217] N. Gupta and R. Hauser, "Kalman filtering with equality and inequality state constraints," *arXiv preprint arXiv:0709.2791*, 2007.

[218] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, 2002.

[219] F. Schoen, "A wide class of test functions for global optimization," *Journal of Global Optimization*, vol. 3, no. 2, pp. 133–137, 1993.

[220] Z. Kang, J. Chen, and B. Wang, "Global Registration of Subway Tunnel Point Clouds Using an Augmented Extended Kalman Filter and Central-Axis Constraint," *PloS one*, vol. 10, no. 5, 2015.

[221] S. D. Billings, *Probabilistic Feature-Based Registration for Interventional Medicine*. PhD thesis, Johns Hopkins University, 2015.

[222] 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.

[223] D. Glozman, M. Shoham, and A. Fischer, "A surface-matching technique for robot-assisted registration," *computer aided surgery*, vol. 6, no. 5, pp. 259–269, 2001.

[224] 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*, pp. 762–769, 2001.

[225] M. Pauly, M. Gross, and L. P. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proceedings of the conference on Visualization'02*, pp. 163–170, IEEE Computer Society, 2002.

[226] P. Cignoni, C. Montani, and R. Scopigno, "A comparison of mesh simplification algorithms," *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, 1998.

[227] 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*, pp. 209–216, ACM Press/Addison-Wesley Publishing Co., 1997.

[228] AIM@SHAPE model repository, "Fertility point cloud scan," *http://visionair.ge.imati.cnr.it/ontologies/shapes/releases.jsp*, 2004.

[229] 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*, pp. 1828–1835, IEEE, 2013.

[230] 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.

[231] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, June 2001.

[232] K. Pulli and M. Pietikäinen, "Range image segmentation based on decomposition of surface normals," in *Proceedings of the Scandinavian conference on image analysis*, vol. 2, pp. 893–893, Citeseer, 1993.

[233] G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP registration using invariant features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90–102, 2002.

[234] C. Schutz, T. Jost, and H. Hugli, "Multi-feature matching algorithm for free-form 3D surface registration," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 2, pp. 982–984, IEEE, 1998.

[235] T. Zinßer, J. Schmidt, and H. Niemann, "A refined ICP algorithm for robust 3-D correspondence estimation," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. II–695, IEEE, 2003.

[236] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299–309, 2005.

[237] L. Armesto, J. Minguez, and L. Montesano, "A generalization of the metric-based iterative closest point technique for 3D scan matching," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1367–1372, IEEE, 2010.

[238] C. Olsson, F. Kahl, and M. Oskarsson, "The registration problem revisited: Optimal solutions from points, lines and planes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1206–1213, IEEE, 2006.

[239] W. M. Wells III, "Statistical approaches to feature-based object recognition," *International Journal of Computer Vision*, vol. 21, no. 1-2, pp. 63–98, 1997.

[240] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Mathematical Methods in Biomedical Image Analysis, 2000. Proceedings. IEEE Workshop on*, pp. 190–197, IEEE, 2000.

[241] S. Granger, X. Pennec, and A. Roche, "Rigid point-surface registration using an EM variant of ICP for computer guided oral implantology," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 752–761, Springer, 2001.

[242] C. Studholme, D. L. Hill, and D. J. Hawkes, "Automated 3D Registration of Truncated MR and CT Images of the Head," in *BMVC*, vol. 95, pp. 27–36, Citeseer, 1995.

[243] J. Chen, Z. Ding, and F. Yuan, "Theoretical uncertainty evaluation of stereo reconstruction," in *The 2nd International Conference on Bioinformatics and Biomedical Engineering*, pp. 2378–2381, IEEE, 2008.

[244] L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H.-P. Meinzer, and J. M. Fitzpatrick, "Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 8, pp. 1520–1532, 2012.

[245] 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.

[246] J. Havlík and O. Straka, "Performance evaluation of iterated extended Kalman filter with variable step-length," in *Journal of Physics: Conference Series*, vol. 659, p. 012022, IOP Publishing, 2015.

[247] B. M. Bell and F. W. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[248] J. Palep, "Robotic assisted minimally invasive surgery," *Journal of Minimal Access Surgery*, vol. 5, pp. 1–7, Jan 2009.

[249] A. Garg, S. Sen, R. Kapadia, Y. Jen, S. McKinley, L. Miller, and K. Goldberg, "Tumor localization using automated palpation with gaussian process adaptive sampling," in *CASE*, pp. 194–200, IEEE, 2016.

[250] K. Patath, R. A. Srivatsan, N. Zevallos, and H. Choset, "Dynamic Texture Mapping of 3D models for Stiffness Map Visualization," in *Workshop on Medical Imaging, IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[251] P. Puangmali, K. Althoefer, L. D. Seneviratne, D. Murphy, and P. Dasgupta, "State-of-the-art in force and tactile sensing for minimally invasive surgery," *IEEE Sensors Journal*, vol. 8, no. 4, pp. 371–381, 2008.

[252] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "A review of tactile sensing technologies with applications in biomedical engineering," *Sensors and Actuators A: physical*, vol. 179, pp. 17–31, 2012.

[253] J. H. Shuhaiber, "Augmented reality in surgery," *Archives of surgery*, vol. 139, no. 2, pp. 170–174, 2004.

[254] D. Teber, S. Guven, T. Simpfendörfer, M. Baumhauer, E. O. Güven, F. Yencilek, A. S. Gözen, and J. Rassweiler, "Augmented reality: a new tool to improve surgical accuracy during laparoscopic partial nephrectomy? Preliminary in vitro and in vivo results," *European urology*, vol. 56, no. 2, pp. 332–338, 2009.

[255] L.-M. Su, B. P. Vagvolgyi, R. Agarwal, C. E. Reiley, R. H. Taylor, and G. D. Hager, "Augmented reality during robot-assisted laparoscopic partial nephrectomy: toward real-time 3D-CT to stereoscopic video registration," *Urology*, vol. 73, no. 4, pp. 896–900, 2009.

[256] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, "Tissue property estimation and graphical display for teleoperated robot-assisted surgery," in *ICRA*, pp. 4239–4245, 2009.

[257] S. McKinley, A. Garg, S. Sen, R. Kapadia, A. Murali, K. Nichols, S. Lim, S. Patil, P. Abbeel, A. M. Okamura, *et al.*, "A single-use haptic palpation probe for locating subcutaneous blood vessels in robot-assisted minimally invasive surgery," in *CASE*, pp. 1151–1158, IEEE, 2015.

[258] L. Wang, Z. Chen, P. Chalasani, J. Pile, P. Kazanzides, R. H. Taylor, and N. Simaan, "Updating virtual fixtures from exploration data in force-controlled model-based telemanipulation," in *ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V05AT07A031–V05AT07A031, American Society of Mechanical Engineers, 2016.

[259] A. S. Naidu, M. D. Naish, and R. V. Patel, "A Breakthrough in Tumor Localization," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/17, 2017.

[260] A. L. Trejos, J. Jayender, M. Perri, M. D. Naish, R. V. Patel, and R. Malthaner, "Robot-assisted tactile sensing for minimally invasive tumor localization," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1118–1133, 2009.

[261] K. A. Nichols and A. M. Okamura, "Methods to segment hard inclusions in soft tissue during autonomous robotic palpation," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 344–354, 2015.

[262] R. E. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, no. 1, pp. 71–87, 2013.

[263] R. D. Howe, W. J. Peine, D. Kantarinis, and J. S. Son, "Remote palpation technology," *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, no. 3, pp. 318–323, 1995.

[264] J. Marescaux, F. Rubino, M. Arenas, D. Mutter, and L. Soler, "Augmented-reality–assisted laparoscopic adrenalectomy," *Jama*, vol. 292, no. 18, pp. 2211–2215, 2004.

[265] N. Haouchine, J. Dequidt, I. Peterlik, E. Kerrien, M.-O. Berger, and S. Cotin, "Towards an accurate tracking of liver tumors for augmented reality in robotic assisted surgery," in *ICRA*, pp. 4121–4126, 2014.

[266] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum a posteriori estimation for binary images," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 271–279, 1989.

[267] A. Gotovos, N. Casati, G. Hitz, and A. Krause, "Active learning for level set estimation," in *IJCAI*, pp. 1344–1350, 2013.

[268] E. M. Mikhail, J. S. Bethel, and J. C. McGlone, "Introduction to modern photogrammetry," *New York*, 2001.

[269] Y. M. Kim, C. Theobalt, J. Diebel, J. Kosecka, B. Miscusik, and S. Thrun, "Multi-view image and tof sensor fusion for dense 3d reconstruction," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1542–1549, IEEE, 2009.

[270] P. L. Houtekamer and H. L. Mitchell, "Ensemble kalman filtering," *Quarterly Journal of the Royal Meteorological Society*, vol. 131, no. 613, pp. 3269–3289, 2005.

[271] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[272] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Lessons from the amazon picking challenge," *arXiv preprint arXiv:1601.05484*, 2016.

[273] Q.-X. Huang, S. Flöry, N. Gelfand, M. Hofer, and H. Pottmann, "Reassembling fractured objects by geometric matching," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 569–578, 2006.

[274] Chintalapani, T. G, and Russel, "Construction of Multi-Component Statistical Shape Models of Bone Anatomy," in *(Abstract in Proceedings), in Computer Assisted Orthopaedic Surgery (CAOS)*, 2009.

[275] G. Chintalapani, *Statistical atlases of bone anatomy and their applications*. The Johns Hopkins University, 2010.

[276] S. Billings, H. J. Kang, A. Cheng, E. Boctor, P. Kazanzides, and R. Taylor, "Minimally invasive registration for computer-assisted orthopedic surgery: combining tracked ultrasound and bone surface points via the p-imlop algorithm," *International journal of computer assisted radiology and surgery*, vol. 10, no. 6, pp. 761–771, 2015.

[277] M. Parry and E. Fischbach, "Probability distribution of distance in a uniform ellipsoid: Theory and applications to physics," *Journal of Mathematical Physics*, vol. 41, no. 4, pp. 2417–2433, 2000.

[278] H. Tjaden, U. Schwanecke, and E. Schömer, "Real-time monocular segmentation and pose tracking of multiple objects," in *European conference on computer vision*, pp. 423–438, Springer, 2016.

[279] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 199–208, IEEE, 2017.

[280] L. Guan, T. Yu, P. Tu, and S.-N. Lim, "Simultaneous image segmentation and 3d plane fitting for rgb-d sensorsan iterative framework," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp. 49–56, IEEE, 2012.

Acronyms

A-ICP anisotropic ICP 247, 248, 264, 266, 268

ALSE active level set estimation 289, 290

bDQF batch dual quaternion-based filter 27, 229–231, 233, 236, 238

BF Bingham distribution-based filter 24, 162, 182, 184–187, 189, 193–195, 197, 198, 246

CMU complementary model update 3, 4, 75, 76, 78, 82, 84, 89–91, 96, 100, 279

CSA complementary situational awareness 89, 278

CT computed tomography 17, 40, 50, 130, 131, 225, 275

DICP deformation compensated ICP 88, 91, 100

DOFs degrees of freedom 131, 139, 152, 157

DQF dual quaternion-based filter 19, 20, 104, 115–117, 119, 121–126, 165, 182–184, 191, 214, 215, 227–229, 232, 238

dSPR deterministic sparse point registration 27, 28, 227, 232, 233, 235–238, 240–242

dVRK daVinci research kit 29, 90, 276–278, 284, 290, 294, 298

EKF extended Kalman filter 20, 38, 39, 42, 43, 73, 105, 106, 117, 119, 121–124, 136, 162, 165, 182–184, 206, 214, 248, 265, 268

EM expectation maximization 245, 247, 251

EM-ICP expectation maximization ICP 247, 263, 266

FSR force sensitive resistor 280, 290

G-ICP generalized ICP 246, 247, 264, 266, 268

GA genetic algorithms 204, 209, 214, 215, 217, 218, 251

Go-ICP globally optimal ICP 39, 191, 214, 259, 302, 308

GP Gaussian process 287, 289

GPR Gaussian process regression 30, 88, 278, 288

GTLS-ICP generalized total least squares ICP 246, 247, 264, 266, 268

HKA heuristic Kalman algorithm 26, 202, 208, 210, 212, 213, 215, 217

ICP iterative closest point 24, 27, 36, 37, 45, 57, 77, 87, 88, 91, 100, 105, 163, 177, 191, 193, 194, 214, 215, 217, 218, 227, 230–233, 236, 238, 247, 248, 262–264, 266, 268, 279, 305, 306, 311

IEKF iterative extended Kalman filter 39, 77, 78, 103, 106, 155, 166, 248, 265

IMLOP iterative most likely oriented point 36, 164, 191, 192

IMLP iterative most likely point 238, 246–249, 264, 266, 268, 309, 310

MAP maximum *a posteriori* 37, 42, 43, 248, 249, 253, 267, 306, 309

MIS minimally invasive surgery 75, 76, 100, 130

MLE maximum likelihood estimate 43, 248, 249, 253, 255, 262–264, 267, 306, 309

MRI magnetic resonance imaging 40, 50, 130, 131, 225, 275

MSBP multiple start branch and prune filter 25–27, 34, 48, 49, 201, 202, 208–219, 221–223

PD-tree principal direction tree 163, 179, 191

PDF probability density function 20, 37, 68, 69, 126, 129, 130, 164, 171

PSO particle swarm optimization 204, 209, 251

pSPR probabilistic sparse point registration 28, 227, 233, 236–238, 240–242

RD recursive decomposition 204

RMS root mean square 24, 88, 122, 184, 186, 187, 189, 193–195, 197, 219, 221, 233, 234, 236, 238, 241, 283, 284, 297

ROI region of interest 30, 292, 293

ROS robot operating system 90, 283

SA simulated annealing 204, 209, 213, 215, 217, 218

SCAR simultaneous compliance and registration estimation 11, 40, 75, 78, 101, 130, 142, 152, 158, 159

SPR sparse point registration 48, 49, 225–227, 230, 232, 239, 241

UKF unscented Kalman filter 20, 21, 38, 39, 43, 73, 84, 105, 106, 109, 117, 119, 121–124, 148, 149, 151, 154–156, 162, 165, 166, 182–184, 206, 214, 233, 238, 248, 265, 268

UPF unscented particle filter 233, 236, 238

US ultrasound 50, 77, 130, 131