LINEAR OBJECT REGISTRATION FOR IMAGE-GUIDED INTERVENTIONS

by

Matthew S. Holden

A thesis submitted to the School of Computing

In conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

(August, 2014)

Abstract

Purpose: Image-guided interventions rely on registration of images, models, and surgical tools into a common navigation space. Point-set registration is commonly used to perform this registration, but requires well-defined landmark points to be present on these tools. In this thesis, a generalization of point-set registration is proposed to simultaneously register point, line, and plane landmarks present on surgical tools. This facilitates registration when point-set registration is not feasible. **Methods:** The proposed algorithm first determines correspondences between points, lines, and planes in the coordinate systems using a set of "reference" landmarks, then calculates invariant features in each coordinate frame for an initial registration, and finally optimizes the registration iteratively. Several forms of validation are investigated: registration of simulated data with a known ground-truth registration, phantom registration using a tracked stylus for registration to a model or a volume, and volume registration of a reconstructed ultrasound volume to a model. Validation accuracy is determined by comparison to a known ground-truth registration or using registration quality metrics such as target registration error. **Results:** For the simulated data experiments, the linear object registration was sufficiently close to the ground-truth registration in all cases given the level of noise. For real registration experiments, in all instances where accurate point-set registration was possible, the linear object registration was equally as accurate, and the difference between the two registrations was less than the fiducial localization error. When accurate point-set registration was not possible, the linear object registration was observed to be more accurate and more precise than point-set registration using approximate landmarks. **Conclusion:** The proposed linear object registration algorithm is a viable alternative when point-set registration cannot be performed. The algorithm has been developed as an open-source registration tool for practical use as a module for the 3D Slicer platform.

Statement of Co-Authorship

The work described in this thesis was completed under the supervision of Dr. Gabor Fichtinger in the Laboratory for Percutaneous Surgery, School of Computing, Queen's University.

Much of the work presented in this thesis has been previously presented at the 12th Imaging Network Ontario Symposium (Matthew S. Holden, Gabor Fichtinger, "Linear Object Registration: A Registration Algorithm Using Points, Lines, and Planes", 12th Imaging Network Ontario Symposium, Mar 2014) and has been accepted for presentation and publication in proceedings at the Workshop on Augmented Environments for Computer Assisted Interventions (Matthew S. Holden, Gabor Fichtinger, "Linear Object Registration of Interventional Tools", Augmented Environments for Computer Assisted Interventions, Sep 2014).

Acknowledgements

First, I am most grateful to my supervisor, Dr. Gabor Fichtinger, for his support during my entire time in the Laboratory for Percutaneous Surgery and while working on this thesis. His mentorship has been invaluable, and he has promoted my growth in both research and academics. I thank him for taking me on as a student and providing me with such wonderful opportunities.

I would like to express my gratitude to my colleagues and friends from the Laboratory for Percutaneous Surgery. It has been a great atmosphere in which to learn and grow thanks to them; they have made the experience so much more enjoyable. Special thanks to Dr. Tamas Ungi and Dr. Andras Lasso for their advice and guidance.

I was supported financially by the NSERC Canada Graduate Scholarship (2012-2013) and the Ontario Graduate Scholarship (2013-2014). Dr. Gabor Fichtinger was supported as a Cancer Care Ontario Research Chair in Cancer Imaging.

Finally, I would like to thank my family and friends for encouraging me to pursue research in whatever field I wished and for their unwavering love and support throughout my studies. I thank them for always being there when I needed them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3D 3 Dimensions/Dimensional

CT Computed Tomography

DOF Degrees Of Freedom

ICP Iterative Closest Point

FLE Fiducial Localization Error

FRE Fiducial Registration Error

MRI Magnetic Resonance Imaging

PRA Point Reconstruction Accuracy

TRE Target Registration Error

Chapter 1

INTRODUCTION AND BACKGROUND

1.1 Clinical Motivation

Registration, the process of expressing multiple sets of data in the same coordinate frame, is mandatory for all computer-assisted interventions. This process is what allows medical staff to visualize multiple components of an intervention in a common navigation space. For example, in image-guided surgeries, images produced by multiple modalities must be registered and fused to provide medical staff with information from all imaging modalities.

One driving clinical application is phantom registration for simulation training. In simulation training, a medical trainee performs an intervention on a "phantom" model which simulates relevant human anatomy. It has been demonstrated that using an augmented reality display to show the trainee anatomy that cannot be seen by the naked eye improves the efficacy of training [1]. To show this augmented reality display, the interventional tools (e.g. needle) and the phantom must be registered to a common navigation space. This is achieved by phantom registration.

Another motivating application is fiducial frame registration for image-guided robotic procedures. In image-guided robotic procedures, pre-operative images of the patient are used to plan the procedure. To successfully perform the procedure, a rigid fiducial frame is attached to the patient. The pre-operative image of the fiducial frame must be registered to the perioperative fiducial frame pose to execute the plan in the correct coordinate system.

For each of the two above described systems, registration represents a necessary step in the procedure. In this paper, a registration algorithm is proposed to address such registration problems, and the algorithm's applicability to other registration problems is also demonstrated.

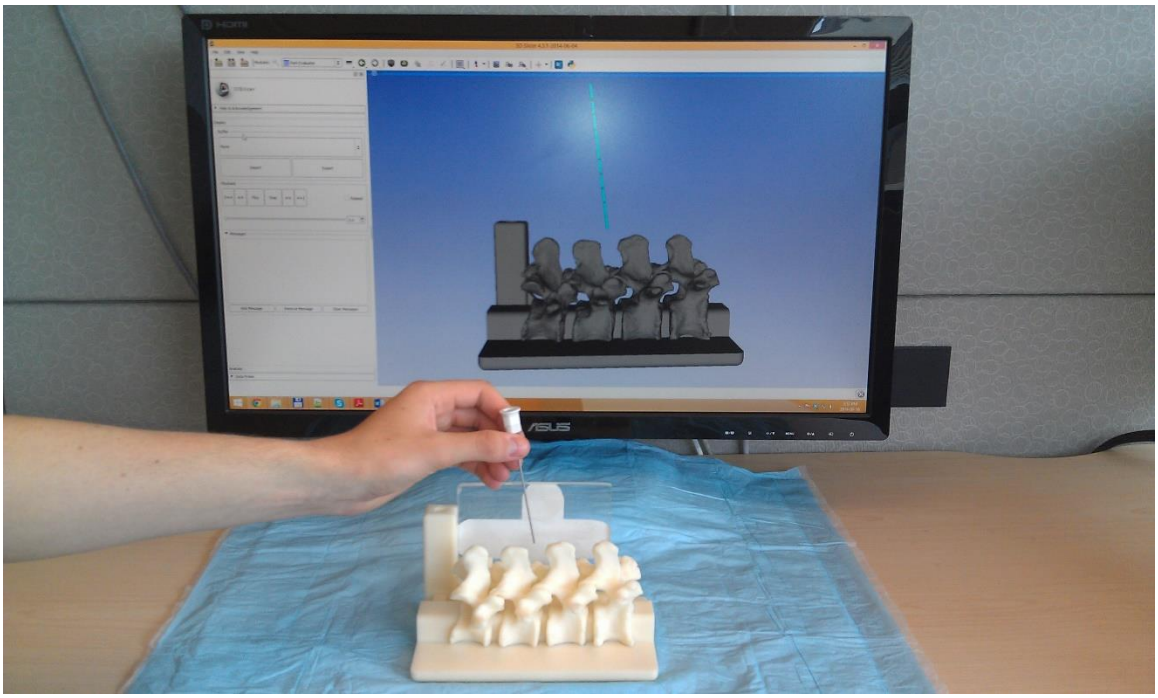

Figure 1.1: Illustration of tracked needle and lumbar spine phantom used for simulation training registered to a common navigation space.

Most computer-assisted interventions use landmarks points which are present on the surgical tool or phantom for registration. These landmark points are usually identified manually in both the tool tracker coordinate system and the navigation coordinate system. The registration can be calculated using a closed-form solution [2], [3]. In many applications, however, the surgical tools or phantoms which must be registered do not have landmark points, but have points, lines, or planes due to engineering constraints, see for example, [4], [5], and [6]. These linear features could be used to perform registration instead.

To this end, a solution for registration using points, lines, and planes (collectively referred to as "linear objects") is sought. The algorithm must satisfy the following requirements to be usable for the desired clinical applications:

- Find a solution for any set of linear objects which uniquely defines a registration
- Find a solution when the correspondence between linear objects is unknown
- Find a solution when the sets of linear objects are of different sizes
- Approach the globally optimal solution as the noise amplitude approaches zero
- Run in polynomial time

Here, an algorithm which satisfies these requirements is presented and integrated as a practical registration tool in an image-guided therapy software system.

1.2 Point-Set Registration

Many modern registration algorithms are based on the closed-form solution to point-set registration with known correspondence. This problem is a specific application of the well-known orthogonal Procrustes problem, which was solved in 1966 by Schonemann [7]. The solution to the point-set registration problem was first introduced by Arun *et al.* [2] and Horn [3] in 1987. Point-set registration will act as the foundation upon which all algorithms presented here will be built. Below, the methods proposed by Arun *et al.* [2] and Horn [3] are described in the context of the following registration problem: register a point-set $\vec{x}_i$ to a corresponding point-set $\vec{y}_i$ by a homogeneous transformation matrix T such that T minimizes fiducial registration error (FRE).

$$FRE^2 = \frac{1}{N}\sum_{i=1}^{N} |T\vec{x}_i - \vec{y}_i|^2$$

1.2.1 Point-Set Registration Algorithm (Arun *et al.*)

Step 1: Transform each point such that the centroid of the point-set is at the origin.

$$\vec{x}_i' = \vec{x}_i - \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

$$\vec{y}_i' = \vec{y}_i - \frac{1}{N}\sum_{i=1}^{N} \vec{y}_i$$

Step 2: Assemble each point-set into a matrix.

$$X = \begin{bmatrix} \vec{x}_1{}'^T \\ \vdots \\ \vec{x}_N{}'^T \end{bmatrix}$$

$$Y = \begin{bmatrix} \vec{y}_1{}'^T \\ \vdots \\ \vec{y}_N{}'^T \end{bmatrix}$$

Step 3: Calculate the 3-by-3 matrix H.

$$H = X^T Y$$

Step 4: Calculate the singular value decomposition of H.

$$H = U\Lambda V^T$$

Step 5: Calculate the rotation R. If $\det(R) = +1$, then R is a rotation. If $det(R) = -1$, then R is a reflection and the algorithm fails.

$$R = VU^T$$

Step 6: Calculate the translation $\vec{d}$.

$$\vec{d} = \left(\frac{1}{N}\sum_{i=1}^{N} \vec{y}_i\right) - R\left(\frac{1}{N}\sum_{i=1}^{N} \vec{x}_i\right)$$

Step 7: Compose the homogeneous transformation matrix T.

$$T = \begin{bmatrix} R & \vec{d} \\ \vec{0} & 1 \end{bmatrix}$$

1.2.2 Point-Set Registration Algorithm (Horn)

Step 1: Transform each point such that the centroid of the point-set is at the origin.

$$\vec{x}_i' = \vec{x}_i - \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

$$\vec{y}_i' = \vec{y}_i - \frac{1}{N}\sum_{i=1}^{N} \vec{y}_i$$

Step 2: Assemble each point-set into a matrix.

$$X = \begin{bmatrix} \vec{x}_1{}'^T \\ \vdots \\ \vec{x}_N{}'^T \end{bmatrix}$$

$$Y = \begin{bmatrix} \vec{y}_1{}'^T \\ \vdots \\ \vec{y}_N{}'^T \end{bmatrix}$$

Step 3: Calculate the 3-by-3 matrix M.

$$M = X^T Y$$

Step 4: Form the matrix N using the elements of M.

$$N = \begin{bmatrix} M_{1,1} + M_{2,2} + M_{3,3} & M_{2,3} - M_{3,2} & M_{3,1} - M_{1,3} & M_{1,2} - M_{2,1} \\ M_{2,3} - M_{3,2} & M_{1,1} - M_{2,2} - M_{3,3} & M_{1,2} + M_{2,1} & M_{3,1} + M_{1,3} \\ M_{3,1} - M_{1,3} & M_{1,2} + M_{2,1} & -M_{1,1} + M_{2,2} - M_{3,3} & M_{2,3} + M_{3,2} \\ M_{1,2} - M_{2,1} & M_{3,1} + M_{1,3} & M_{2,3} + M_{3,2} & -M_{1,1} - M_{2,2} + M_{3,3} \end{bmatrix}$$

Step 5: Find the largest positive eigenvalue λ^* of N and its corresponding eigenvector $\vec{v}^*$.

Step 6: The rotation R is given in quaternion notation by normalizing $\vec{v}^*$.

Step 7: Calculate the translation $\vec{d}$.

$$\vec{d} = \left(\frac{1}{N}\sum_{i=1}^{N} \vec{y}_i\right) - R\left(\frac{1}{N}\sum_{i=1}^{N} \vec{x}_i\right)$$

Step 8: Compose the homogeneous transformation matrix T.

$$T = \begin{bmatrix} R & \vec{d} \\ \vec{0} & 1 \end{bmatrix}$$

1.2.3 Theoretical Considerations for Point-Set Registration

The objective of point-set registration algorithms is to reduce the fiducial registration error (FRE), or, more generally the feature registration error, as described in Section 1.2.

Fitzpatrick *et al.* [8] provide an expression for the expected fiducial registration error given the fiducial localization error (FLE), assuming a zero-mean 3D Gaussian distribution of noise.

$$\langle FRE^2 \rangle = \left(1 - \frac{2}{N}\right) \langle FLE^2 \rangle$$

Their results show that a large fiducial registration error indicates a poor registration result, but more importantly, they analytically show that a small fiducial registration error does not preclude a poor registration result. Rather, they show that target registration error (TRE) is a much more reliable metric for determining the quality of a registration result. Supposing a set of target points $\vec{x}_A$ is collected in coordinate frame A and a corresponding set of target points $\vec{x}_B$ is collected in coordinate frame B, the target registration error is given by the expression below.

$$TRE = \frac{1}{N} \sum_{i=1}^{N} \left| \vec{x}_{B,i} - T_{A \to B} \vec{x}_{A,i} \right|$$

They show that target registration error is dependent on the number and configuration of fiducials. In particular, the target registration error changes in proportion to $1/\sqrt{N}$, and is proportional to the quotient of the distance from to the centroid and the size of the fiducial configuration.

Furthermore, these results have been extended to show that if the fiducial localization error is normally distributed with zero mean, then target registration error will be distributed likewise and its variance will increase with fiducial localization error [9]. Additionally, several recent works have demonstrated how the target registration error distribution and covariance is affected by inhomogeneous and anisotropic localization error [10], [11], [12]. In the presence of bias in fiducial localization error, Moghari and Abolmaesumi [13] show that the fiducial registration is unaffected, but the target registration error is increased. Their results agree with work from Fitzpatrick [14], which demonstrates that target registration error is uncorrelated with fiducial registration error. Again, this demonstrates that fiducial registration error is an unreliable measure of registration quality.

In summary, the previous works theoretically and experimentally show that target registration error is a better measure of registration quality than fiducial registration error. They provide further results indicating the importance of selecting target points from the entirety of the region of interest. These results motivate the use of target registration error the measure used to evaluate registration quality throughout this paper.

1.3 Point-Set Registration with Unknown Correspondence

In many practical situations, the correspondence between the two point-sets in unknown. That is, there exist point-sets $\vec{x}_i$ and $\vec{y}_i$ which must be registered, but it is possible that the point $\vec{x}_i$ corresponds to the point $\vec{y}_j$ for which $i \neq j$ and possibly $N_x \neq N_y$. In this case, the registration problem becomes: register a point-set $\vec{x}_i$ to a point-set $\vec{y}_i$, with a matching function φ: $1 \dots N_x \rightarrow 1 \dots N_y$, by a homogeneous transformation matrix T such that φ and T minimize the objective function $FRE^2 = \frac{1}{N}\sum_{i=1}^{N}\left|T\vec{x}_i - \vec{y}_{\varphi(i)}\right|^2$. There exists two main approaches for solving these types of registration problem: iterative methods, which iteratively improve the registration and correspondence simultaneously, and feature extraction methods, which attempt to match features in the two point-sets to determine correspondence. Iterative methods have the drawback that they require a sufficient initial guess to converge to the global optimum. Feature extraction methods can be problematic because they require robust feature classification and extraction. Often feature extraction methods or heuristics are used as initial guesses to iterative methods.

In the subsequent sections, several different iterative and feature extraction approaches are discussed and examined. The following examples are not intended to be an exhaustive list of all approaches, but rather, they are intended to illustrate a variety of techniques from the literature.

1.3.1 Iterative Closest Point

The most commonly used method for point-set registration with unknown correspondence is the iterative closest point (ICP) method proposed by Besl and McKay [15] in 1992, described below.

Step 1: Determine an initial guess of the transformation T.

Step 2: For each point $T\vec{x}_i$, compute the closest point $\vec{y}_{\hat{\varphi}(i)}$.

Step 3: Using the current estimated correspondence function, $\hat{\varphi}$, compute the registration T between $\vec{x}_i$ and $\vec{y}_{\hat{\varphi}(i)}$ using the closed-form solution to point-set registration with known correspondence.

Step 4: Repeat steps 2-3, until the convergence criterion is met: $FRE^2{}_k - FRE^2{}_{k+1} < \delta$, where $\delta > 0$ is a preset threshold and k indexes the iteration.

1.3.2 Other Iterative Algorithms

Since the original ICP algorithm proposed by Besl and McKay [15], many modifications and improvements have been introduced to promote convergence to the global optimum. Perhaps the most well-known variant is the point-set to surface registration method proposed by Chen and Medioni [16] (though it should be noted that the work from Chen and Medioni [16] was published prior to the work of Besl and McKay [15]), which proceeds similarly to the ICP algorithm, but matches points to surface locations by projecting the points in the direction of the normal vector onto the surface. A review of ICP variants is provided in Rusinkiewicz and Levoy [17].

There exists a multitude of other iterative algorithms, only several the most common such algorithms are reviewed here. One such algorithm is the Robust Point Matching algorithm proposed by Gold *et al.* [18] in 1998. Their algorithm solves the registration part using the coordinate descent method, followed by a fuzzy matching part using their "Softassign" method

(which uses deterministic annealing approaches to make the matching problem continuous) and iterates until the algorithm converges to a solution. In 2007, Sofka *et al.* [19] proposed an approach which uses weighted correspondences based on the local covariance of the data. Their algorithm simultaneous updates the correspondence weights, local covariances, and registration parameters. More recently, in 2010, Myronenko and Song [20] proposed the Coherent Point Drift algorithm, which models one point set as a combination of Gaussian mixture models. The method uses an expectation-maximization algorithm to iteratively find the parameters which best fit these models to the other point-set.

Other algorithms attempt to reduce the space over which must be searched to find a globally optimal solution. For example, Foroughi *et al.* [21] introduced a method to reduce the space which must be searched to achieve convergence to the global optimum. They did this for pelvic bone registration, where they used the symmetry plane that exists for the pelvic bone to limit their search space to one dimension, then applied ICP to converge to an optimum. Li and Hartley [22] extended this notion of reducing the search space by modifying the registration error function to satisfy the Lipschitz continuity condition (i.e. has a bounded first derivative). Then, provided a sufficiently high search resolution, their algorithm is guaranteed to find the global optimum. Fitzgibbon [23] even suggests that using the more generic Levenberg-Marquardt method for optimization in combination with the ICP algorithm may be more likely to converge to a global optimum than the ICP algorithm alone. Additionally, they present a framework for which the error function can readily be modified, and find that using the Huber kernel improves convergence even further.

1.3.3 Feature Extraction Algorithms

Invariant feature extraction methods provide an alternative to iterative algorithms, using these invariant features to determine the point correspondences and then using the known closed-

form solution. These algorithms may also be used to provide initial estimates to iterative algorithms.

Thirion [24] suggests using crest lines, the set of points for which the largest curvature achieves a local maximum, as invariant features to determine extremal points, the extrema on the crest lines. Then, the extremal points are used to find correspondence between point-sets for registration. Xiao *et al.* [25], on the other hand, calculate for each vertex the distribution of surrounding vertices, and match the distributions to find vertex and point correspondences. Sharp *et al.* [26] provide a list of invariant features that can be used to find correspondence for point-set registration: local curvature, second order moments, spherical harmonics, and a global density function. They further provide a framework for integrating these invariant features into an iterative closest point algorithm and provide convergence results for each feature.

Alternatively, if it can be guaranteed that a sufficient number and configuration of features can be found, the features can be used directly for registration. For example, Jian *et al.* [27] propose a method that approximate the point-set as a continuous distribution using Gaussian mixtures models. Then, they propose a method for directly registering these mixture model features to solve the point-set registration problem. Though, it is noted that such direct feature registration methods may have iterative components.

1.4 Overview of Linear Object Registration Algorithms in the Literature

Several methods for registering two sets of points, lines, and planes have been previously proposed. None of the works found in the literature search, however, present a solution that satisfies all of the outlined requirements for the desired clinical applications.

1.4.1 Fiducial Frame Registration

In many robot-assisted surgeries, fiducial frames are affixed to the patient. These frames are subsequently imaged and used to register images of the patient to a surgical robot's coordinate system. Typically, this method of determining the pose of the image with respect to 3D space uses a fiducial frame with point and line features which are visible by cross-sectional image.

Lee *et al.* [28] present several algorithms for registering a cross-sectional image of a generic fiducial frame consisting of line features to 3D space and provide results for the accuracy of each algorithm. Unfortunately, no closed-form solution to the problem is presented and convergence is not guaranteed with their algorithms. Additionally, their framework requires significant extension to work with point and plane features in addition to lines features.

Another approach for fiducial frame registration was developed by Jain *et al.* [29]. Their approach uses a more generalized fiducial frame, which includes point and ellipse features in addition to line features. They present a particular fiducial frame design which boasts desirable convergence properties. Their algorithm in general, however, is not guaranteed to converge for fiducial frames with unknown geometry. Additionally, their algorithm takes advantage of several features of CT imaging which may not be applicable to all imaging modalities or all registration problems.

1.4.2 Registration of Homologous Points, Lines, and Planes (Meyer *et al.*)

In 1995, Meyer *et al.* [30] presented a closed-form solution to the problem of simultaneous registration of points, lines, and planes for multiple medical images from multiple modalities. To this end, denote the two coordinate frames as A and B and for arbitrary coordinate frame X, denote the i^{th} point, line, or plane as $P_{X,i}$, $L_{X,i}$, and $A_{X,i}$, respectively.

Step 1: In each coordinate frame, calculate the centroid of all points $\vec{c}_A$ and $\vec{c}_B$.

Step 2: Translate each coordinate frame such that the centroid is at the origin.

Step 3: For each line in each coordinate frame, calculate a) the projection of the origin onto the line and b) the vector orthogonal to the projection and direction vectors.

Step 4: For each plane in each coordinate frame, calculate the projection of the origin onto the plane.

Step 5: Use the points and the quantities calculated in steps 3-4 for point-set registration with known correspondence.

This algorithm, however, requires known correspondence between points, lines, and planes, and more importantly, fails when only lines and planes are collected or when the centroid lies on a line or plane. Although their algorithm is guaranteed to achieve an optimum, it is unclear how the optimum is analytically relevant to the initial registration problem.

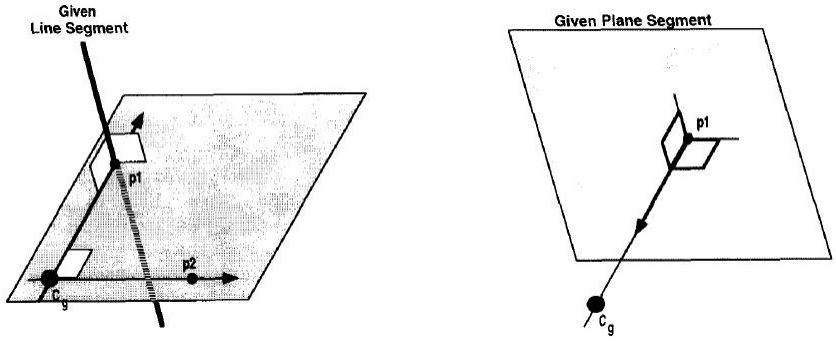

Figure 1.2: Points extracted from lines used for point-set registration (left) and points extracted from planes used for point-set registration (right). Figures from Meyer *et al.* **[30]**.

1.4.3 Points-to-Points, Point-to-Lines, and Points-to-Planes Registration (Olsson *et al.*)

Olsson *et al.* [31] have proposed, in 2006, a solution to registering a set of collected points to points, lines, and planes with unknown correspondence. Their approach is to rewrite the points-to-point, points-to-line, and points-to-plane problems each as a fourth degree polynomial optimization problem in four unknowns of the same form, using a quaternion representation of rotation. The result is a combined non-convex optimization problem in four unknowns. They present a solution which uses a branch and bound approach using the solution of the convex relaxation as a lower bound. The primary drawback with their approach is that it is still an

exponential time complexity algorithm. Further, there is an unjustified assumption of Lipschitz continuity and their approximation algorithm guarantees a solution within ϵ of the optimal value (i.e. fiducial registration error), rather than guaranteeing a solution within ϵ of the optimal solution (i.e. transformation matrix).

1.5 Proposed Algorithm Contribution

The algorithm proposed in this thesis provides a more generic registration framework than point-set registration algorithms (with known or unknown correspondence), however, represents a specific case of algorithms for collected points to surface registration.

The proposed algorithm works in cases when point-set registration does not:

- Works when an arbitrary set of points, lines, and planes is available

The proposed algorithm has advantages over collected points to surface registration:

- Is approximately globally convergent
- Runs in polynomial time

The goal of the algorithm is not necessarily to produce a more accurate solution that point-set registration, nor is it to provide a more general solution to registration that points to surface algorithms. The proposed algorithm is intended to provide a convenient alternative when point-set registration cannot be performed, but points, lines, and planes are available for registration. Further, when such points, lines, and planes are available, it provides better convergence properties than points to surface registration.

Chapter 2

ALGORITHM

2.1 Outline of Algorithm

We propose an algorithm to satisfy the requirements laid out in Section 1.1. The algorithm finds correspondences using external features and then attempts to identify invariant features to determine corresponding points for point-set registration. The workflow is most similar to the work of Meyer *et al.* [30]. Our algorithm, however, attempts to solve a more general problem.

For generality, the proposed algorithm will be described in the context of registration between a coordinate frame A and a coordinate frame B. In each coordinate frame, there should exist a set of linear objects S_A and S_B respectively, where one of S_A or S_B is possibly a permuted subset of the other. The objective of the algorithm is to determine the 4-by-4 homogenous transformation matrix from coordinate frame A to coordinate frame B. Of course, the proposed solution is applicable to any registration problem where points, lines, and planes occur in two coordinate systems.

Points, lines, and planes will not be assumed to be defined in any particular form, though conversion between forms is well-known. The notation $dist(a, b)$ will be used to indicate the Euclidean distance between linear objects a and b, where it will be taken to mean the smallest such distance between a and b in the case that a and b are not both points. The notation $proj_a(b)$ will be used to indicate the projection of b (necessarily a point) onto a linear object a. Finally, the notation LO will be used to indicate the set of all linear objects.

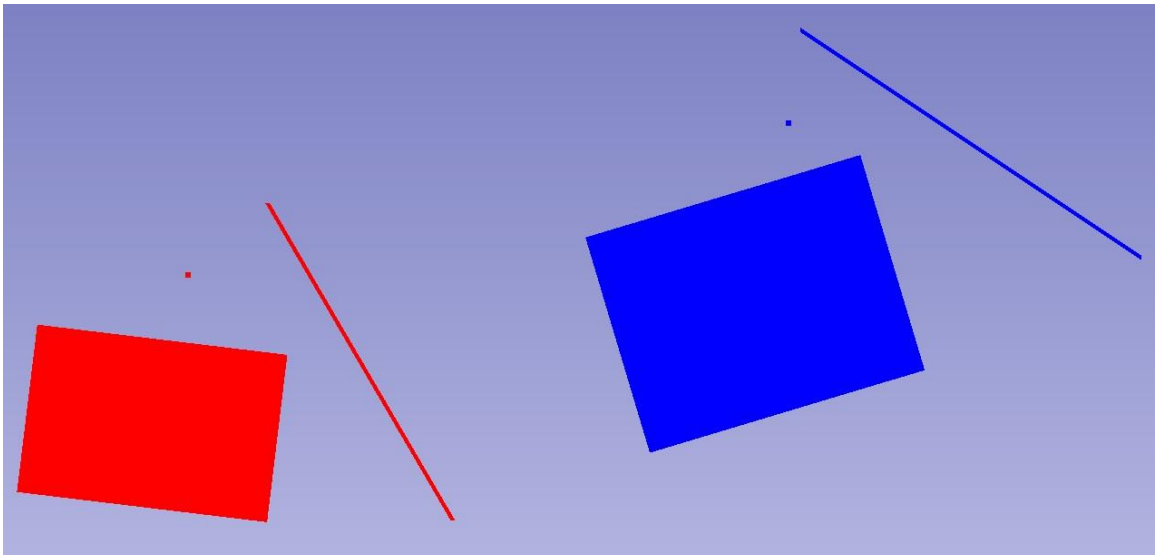

Figure 2.1: Illustration of a set of linear objects ***A*** (red) and a set of linear objects ***B*** (blue).

To satisfy the outlined requirements, the algorithm will: find correspondences between the linear object in the two coordinate frames, then, compute an invariant feature and use that feature for point-set registration with known correspondence. Further, several practical considerations will be examined: how to extract linear objects from various sources of medical data and how to promote convergence given the raw medical data sources.

2.2 Finding Correspondence

The first objective of the algorithm is to find correspondences between linear objects in the set S_A and the set S_B. It cannot be guaranteed that any particular linear object in set S_A corresponds to any particular linear object in set S_B. Furthermore, it cannot be guaranteed that any particular linear object in set S_A even corresponds to any linear object in set S_B (and vice-versa).

Given a ground-truth correspondence function $\varphi: LO \rightarrow LO$, the objective is to find an estimate of that correspondence function $\hat{\varphi}: LO \rightarrow LO$. To this end, consider a function $\rho: LO \rightarrow \mathbb{R}^N$ and some fixed threshold δ such that:

$$\forall a \epsilon S_A, b \epsilon S_B | dist\big(\rho(a), \rho(b)\big) < \delta \iff \varphi(a) = b$$

If such a function ρ and threshold δ can be found, then the estimator $\hat{\varphi}$ can be constructed with the following form:

$$\hat{\varphi}(a) = \begin{cases} \underset{b \epsilon S_B}{\operatorname{argmin}} \big[dist\big(\rho(a), \rho(b)\big)\big] \ \Big(if \ \min_{b \epsilon S_B} \big[dist\big(\rho(a), \rho(b)\big)\big] \ < \delta\Big) \\ undefined \ \Big(if \ \min_{b \epsilon S_B} \big[dist\big(\rho(a), \rho(b)\big)\big] \ > \delta\Big) \end{cases}$$

It can be readily shown that if the function ρ and the threshold δ are chosen appropriately to satisfy the requirement, then this construction of $\hat{\varphi}$ yields $\hat{\varphi} = \varphi$ (see Appendix A for proof).

In practice, it may not be possible find such a function ρ and a fixed threshold δ that works for all possible sets of linear objects S_A and S_B. The above definition for $\hat{\varphi}$ can be relaxed to the following, which offers a better estimate of φ in the case that such a function ρ and fixed threshold δ are not achievable.

The relaxed form of the matching function $\hat{\varphi}$ is determined by solving the linear assignment problem on sets S_A and S_B with cost function $f(a, b) = dist\big(\rho(a), \rho(b)\big)$, where all assignments with weight greater than the fixed threshold δ are rejected. In practice, $\hat{\varphi}$ can be determined in polynomial time by the Kuhn-Munkres assignment algorithm. It can readily be shown that if the function ρ and the fixed threshold δ can be found to satisfy the original condition, then this construction of $\hat{\varphi}$ also yields $\hat{\varphi} = \varphi$ (see Appendix A for proof). But this construction has the added benefit that $\hat{\varphi}$ remains defined even if the condition does not strictly hold.

This, in practice, equates to finding a function ρ that maximizes the following expression:

$$\min_{b \neq \varphi(a) \epsilon S_B} \big[dist\big(\rho(a), \rho(b)\big)\big] - dist\Big(\rho(a), \rho\big(\varphi(a)\big)\Big)$$

In practice, we construct the function ρ by introducing a set of objects called "references", which are points in each coordinate frame with known correspondence, but possibly have significant uncertainty associated with them. Since distances are preserved in rigid

transformations, the distance from a linear object $a \epsilon S_A$ to a reference $a_r \epsilon S_A$ will be the same as the distance from the corresponding linear object $\varphi(a) \epsilon S_B$ to the corresponding reference $\varphi(a_r) \epsilon S_B$. Since the references have known correspondence, it is possible to define the "signature" of a linear object as the vector of distances to references.

$$sig(a) = \begin{bmatrix} dist(a, a_{r1}) \\ \vdots \\ dist(a, a_{rM}) \end{bmatrix}$$

Since the correspondence between references is known, the following most hold (where there is possibly some error due to noise):

$$sig(a) \cong sig\big(\varphi(a)\big)$$

By distributing references sufficiently throughout the region of interest, it can be ensured that both of the following hold (for some fixed threshold δ):

$$dist\left(sig(a), sig\big(\varphi(a)\big)\right) < \delta$$

$$\forall b \neq \varphi(a) \epsilon S_B | dist\big(sig(a), sig(b)\big) > \delta$$

It is observed that this function sig has some of the desired properties of the function ρ that is sought. Thus, the function sig can be used as the function ρ, and the fixed threshold δ can be estimated as the number of references multiplied by the approximate noise associated with localizing the references.

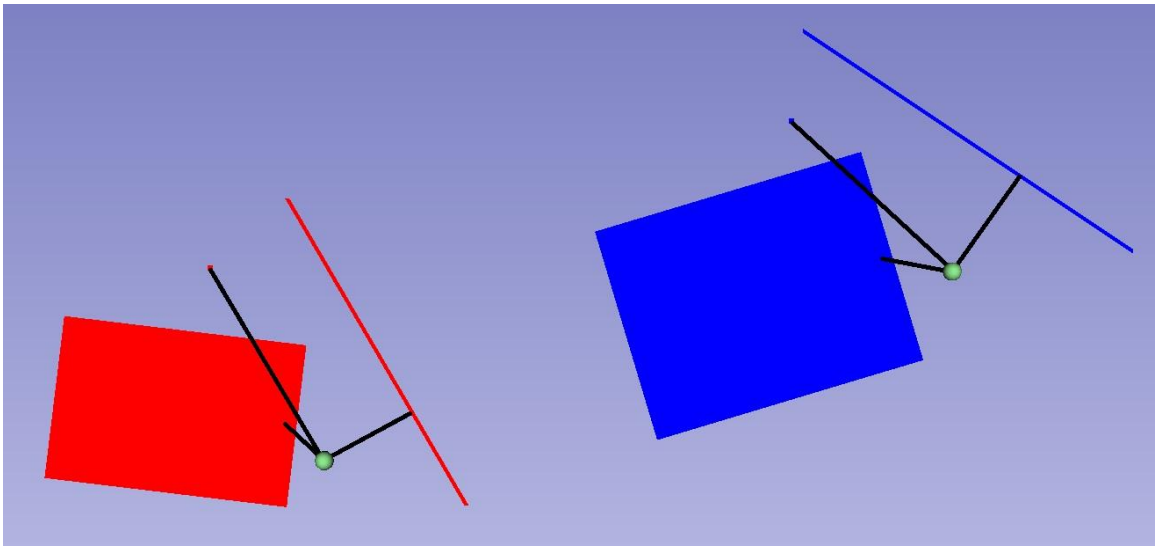

Figure 2.2: Two sets of linear objects with references indicated in green and distances from linear objects to references indicated in black.

The question that remains is how to choose the references to optimize the properties of sig. No theoretical arguments are provided on how to choose references, other than a proof that for any sets of linear objects S_A and S_B, there exists a function sig that satisfies the required condition on ρ (see Appendix A for proof). Some empirical results on how to choose the references, however, are provided in subsequent sections.

It is noted that requiring a set of references does not violate the requirement of not needing any particular set of linear objects. The references are not used explicitly in the registration, and thus, need not be collected with particular accuracy.

2.3 Point-to-Point, Line-to-Line, Plane-to-Plane Registration

Once correspondence between the linear objects has been determined, the linear objects must be registered to calculate the transform from coordinate frame A to coordinate frame B.

2.3.1 Invariance of the Generalized Centroid

Once the correspondence between linear objects has been determined, correspondence between points on the linear objects must be determined. To determine this correspondence, an invariant shall be used. Any invariant $\vec{c}$ must satisfy the property that, given the transformation between two coordinate frames $T_{A\to B}$:

$$\vec{c}_B = T_{A\to B}\vec{c}_A$$

Further, it can be shown that any invariant point has the property that in can reduce a 6DOF rigid registration problem into 3DOF spherical registration problem. This can be shown by decomposing the transformation between the two coordinate frames into a rotational part and a translational part:

$$B = T_{A\to B}A$$

$$B = R_{A\to B}A + \vec{d}_{A\to B}$$

Finally, the invariant can be introduced to remove the need to simultaneously calculate the translation when calculating the rotation:

$$B - \vec{c}_B + \vec{c}_B = R_{A\to B}(A - \vec{c}_A + \vec{c}_A) + \vec{d}_{A\to B}$$

$$B - \vec{c}_B + \vec{c}_B = R_{A\to B}(A - \vec{c}_A) + \left(R_{A\to B}\vec{c}_A + \vec{d}_{A\to B}\right)$$

$$B - \vec{c}_B = R_{A\to B}(A - \vec{c}_A)$$

Following the work of Arun *et al.* [2], Horn [3], and Meyer *et al.* [30], the centroid is used as the invariant feature to both reduce the problem to a spherical registration problem and to determine the correspondences between the points on the linear objects in the two coordinate frames. In this case, however, it cannot be guaranteed that any points exist in either of the sets of linear objects. To this end, a generalization of the centroid is proposed and used for registration.

The centroid of a set of linear objects is defined to be the point (or set of points) which has a minimum sum of squares distances to all linear objects.

$$f = \sum_{Point} \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,1} \right]^2 + \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,2} \right]^2 + \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,3} \right]^2$$

$$+ \sum_{Line} \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,1} \right]^2 + \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,2} \right]^2 + \sum_{Plane} \left[\left(\vec{x} - \vec{b}_i\right)^T \vec{n}_{i,1} \right]^2$$

The objective is to find a point (or set of points) $\vec{x}$ that is a global minimizer of the function f. Here, it is assumed that each point, line, and plane has some point $\vec{b}_i$ that is known to lie on it along with associated normal vectors $\vec{n}_{i,j}$.

This definition is consistent with the usual definition of centroid for a set of points which minimizes the sum of squares distances to all points. The solution is based on the least-squares formulation and is given below (assuming that $\vec{n}$ and $\vec{b}$ are indexed appropriately over a total of m linear objects).

$$\vec{x} = (N^T N)^{-1} N^T B$$

$$N = \begin{bmatrix} \vec{n}_1{}^T \\ \vdots \\ \vec{n}_m{}^T \end{bmatrix}, B = \begin{bmatrix} \vec{b}_1{}^T \vec{n}_1 \\ \vdots \\ \vec{b}_m{}^T \vec{n}_m \end{bmatrix}$$

The centroid for a set of linear objects exhibits several important properties (or variants of properties applicable to the problem at hand): existence, uniqueness, and invariance (see Appendix A for proofs). Most importantly, if the set of linear objects uniquely defines a registration, then the centroid is guaranteed to exist, be unique, and is an invariant of the registration. Further, the generalized centroid exhibits the same robustness properties as the "traditional" centroid, where it is noted that planes have less specificity than lines, which have less specificity than points, due to their dimensionality.

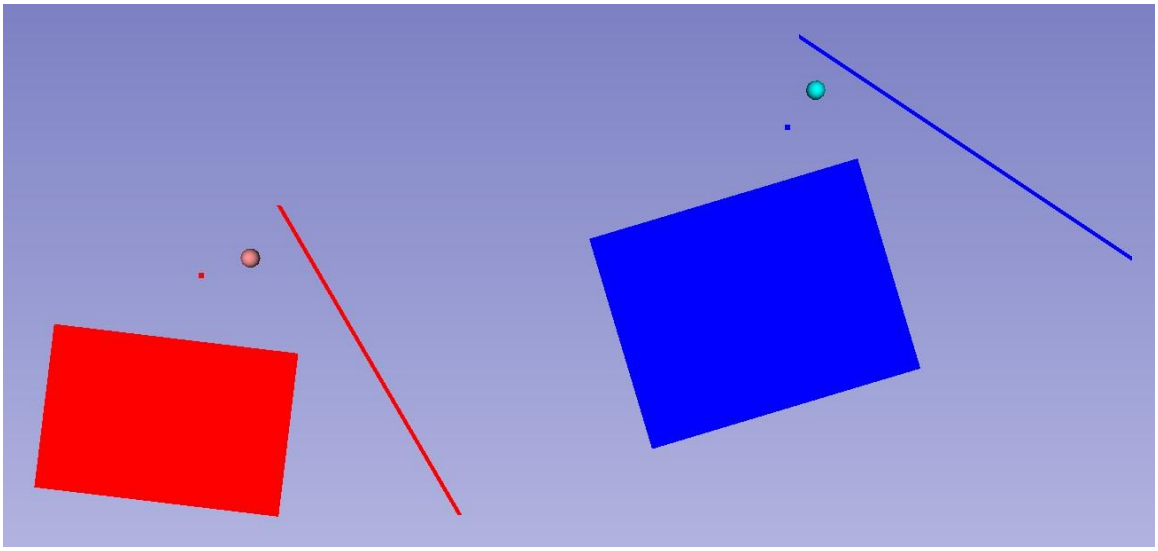

Figure 2.3: Two sets of linear objects, with the linear object centroid of the red set indicated in light red and the linear object centroid of the blue set indicated in light blue.

2.3.2 Corresponding Points on Linear Objects

Once the centroid has been found, it can be used to determine corresponding points on the linear objects in the two coordinate frames. Then, the points with known correspondence can be used for point-set registration with known correspondence, thus, providing a solution to the linear object registration problem. The objective is to find as many points as possible with known correspondence and use all of them to calculate the registration.

First, it is noted that the projection of the centroid onto each linear object is invariant under transformation (see Appendix A for proof) and the correspondence between the projections is known. Thus, these points may be used in calculating the registration (see Appendix A for proof).

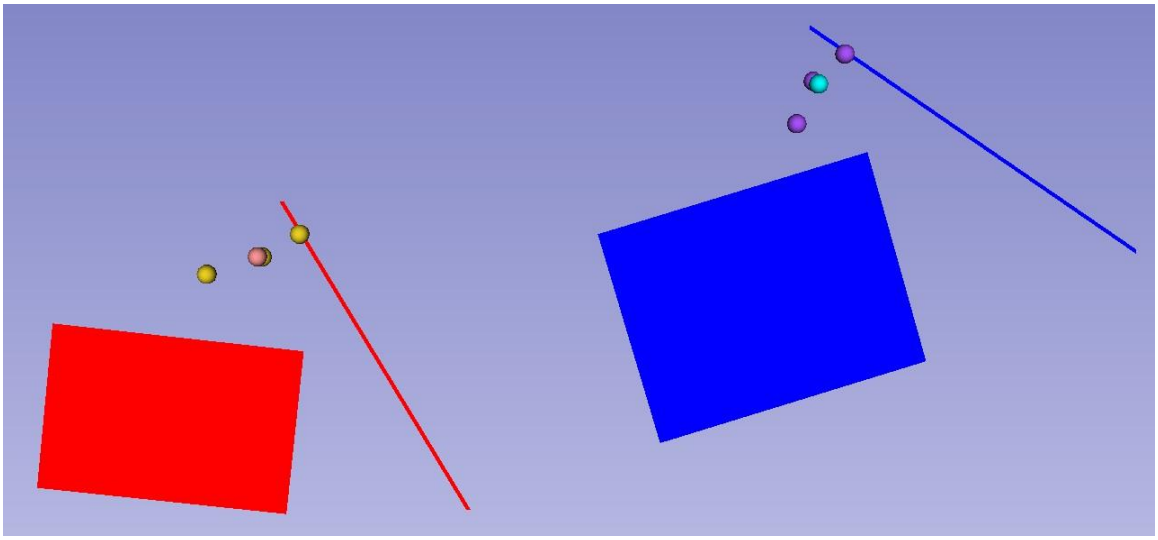

Figure 2.4: Illustration of projections of the linear object centroids onto linear objects of each type. Projections for the red set are indicated in yellow, projections for the blue set are indicated in purple.

These projections, however, are the only points on all types of linear objects that have a unique distance to the centroid. There is ambiguity associated with all other points based solely on distance.

The other values that can be used for point-set registration with known correspondence are the direction vectors associated with lines and the normal vectors associated with planes. These vectors can also be added to the point-set registration with known correspondence problem. The issue arising with direction and normal vectors is that there is ambiguity associated with them. The negative of any direction or normal vector is also a valid direction or normal vector for the same linear object. To resolve these ambiguities, the references introduced to resolve the initial linear object correspondence problem may be reused.

The same matching protocol is used to find corresponding direction and normal vectors. The positive (or negative) direction/normal vector $\vec{v}_A$ may be arbitrarily chosen in coordinate frame A. Then, of the two possible direction/normal vectors $\vec{v}_B$ in coordinate frame B, the one which minimizes the expression below is chosen.

$$dist\big(sig(\vec{v}_A), sig(\vec{v}_B)\big)$$

These direction/normal vectors from the lines and planes can be added to the set of points used for point-set registration with known correspondence.

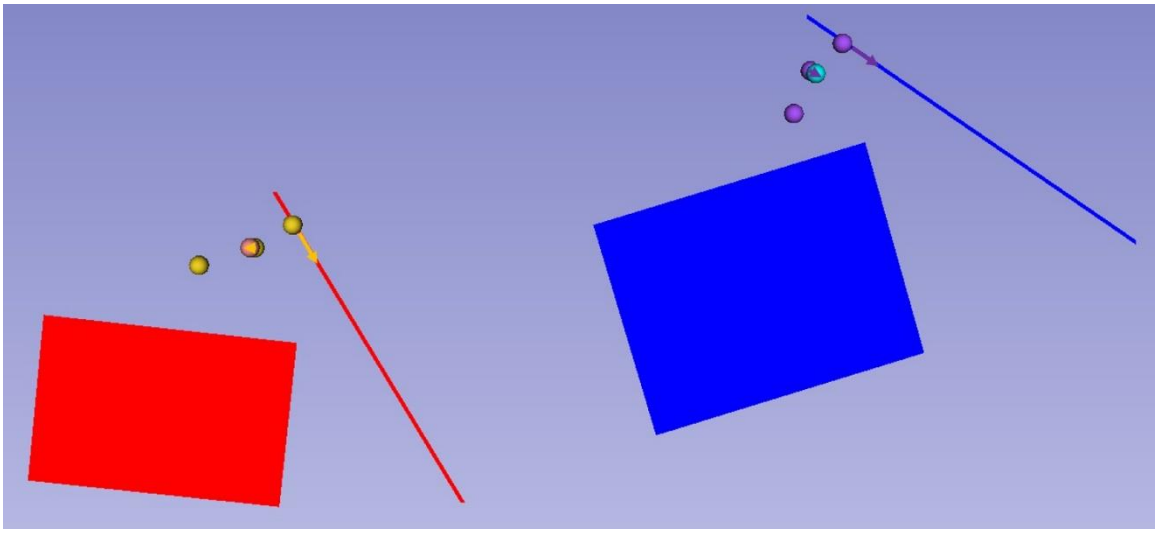

Figure 2.5: Illustration of points and vectors (shown in yellow for the red set of linear objects and shown in purple for the blue set) extracted from the linear object sets used to perform registration.

By combining the projections of the centroid along with the direction/normal vectors associated with lines and planes, point-set registration with known correspondence can be performed. This point-set registration exhibits several desirable properties: if the linear objects define a unique registration then this point-set registration produces a unique solution, and it produces an exact solution to a finite version of the registration problem (see Appendix A for proofs). Thus, this registration is used as the solution to the initial registration problem of points, lines and planes.

2.4 Extracting Linear Objects from Medical Data

In image-guided interventions, there exist three main types from data from which linear objects may be extracted: tracking data, imaging data, and geometrical models. For each source of information, linear objects must be extracted either automatically or semi-automatically.

2.4.1 Extracting Linear Objects from Tracking Data

Tracking data can be produced by a variety of position tracking technologies, including electromagnetic, visible optical, and infrared optical systems. The commonality amongst these technologies is that the tool's trajectory is sampled at a series of points in time. This yields a time-series of position values for the tool. Importantly, this means that the positions are necessarily ordered, which is a property that extraction methods may use to their advantage.

To demarcate the linear objects on a physical object using a tracked tool, the tool must have a tip which can be used to physically probe each linear object. By performing stylus calibration [32], the position of the tooltip can be determined from the position of the tool's sensor. The user must physically slide the tooltip over each line or plane, or pivot the tooltip about each point. Ideally, the user would sample points from the entirety of each linear object, however, this is not required. The sampling may be discontinuous or sparse. Additionally, the probing of each linear object may be explicitly delineated by the user.

Figure 2.6: Photograph of user with a tracked stylus collecting a planar linear object on a lumbar spine phantom.

For a given linear object, the set of collected tooltip positions can be linearly fit using principal components analysis. Supposing X is the set of collected positions $\vec{x}_i$ corresponding to collection on one linear object and X has more elements than some fixed threshold N_E, then the centroid $\vec{c}$ of the positions can be computed.

$$\vec{c} = \frac{1}{N}\sum_{i=1}^{N} \vec{x}_i$$

Subsequently, the principal components analysis is performed by calculating the eigenvalues of the covariance matrix. Supposing the eigenvalues are given by $\lambda_1, \lambda_2, \lambda_3$ and the associated eigenvectors are $\vec{v}_1, \vec{v}_2, \vec{v}_3$, the eigenvectors associated with the eigenvalues with magnitude larger than the squared noise σ^2 of the tool tracking system indicate direction vectors associated with the linear object. Zero such eigenvalues indicate a point, one such eigenvalue indicates a line, two such eigenvalues indicate a plane, and more than two such eigenvalues indicate a poorly defined linear object that is rejected. This allows the user to extract linear objects from physical objects with relative ease.

Of course, tracking data has the property that the order of tooltip positions is known. This order can be used to delineate linear objects automatically in real-time. Every time a new tool position $\vec{x}_i$ is observed, the following algorithm may be executed to automatically delineate linear objects:

1. Add the newly observed position $\vec{x}_i$ to the set X.
2. Determine the dimensionality of the linear object defined by X (using the algorithm described above).
    a. If $\lambda_1, \lambda_2, \lambda_3 > \sigma^2$ (all eigenvalues greater than the noise amplitude) and $|X| \geq N_E$ (a sufficiently large set of points has been collected) then compute the linear object defined by the set $X_T = X \backslash \{\vec{x}_1 \dots \vec{x}_{N_T} \cup \vec{x}_{N-N_T+1} \dots \vec{x}_N\}$, where $N_T < \frac{N_E}{2}$ is a fixed threshold value, then assign $X = \emptyset$.

b. If $\lambda_1, \lambda_2, \lambda_3 > \sigma^2$ (all eigenvalues greater than the noise amplitude) and $|X| < N_E$ (an insufficiently large set of points has been collected), then assign $X = \emptyset$.

3. Repeat until collection is finished.

This algorithm is premised on the notion that once a user stops collecting a linear object, their subsequent motions will be in three dimensions of translational motion. This is a valid assumption for any freehand collection of linear objects. Additionally, it is assumed that any time a user wishes to collect a linear object, they will do so continuously or in several continuous phases. As a tactic to reduce noise, the initial and final N_T positions collected are removed from linear object computation due to possible transient behaviour associated with those positions.

It is also important to note that this algorithm is conducive to being applied in real-time as only a constant number of operations are required each time a position $\vec{x}_i$ is added to the set X. This assumes that the covariance matrix is not fully recalculated every time a position is added, but only updated. Since the dimensionality of the covariance matrix is always 3-by-3, calculating its eigenvalues and eigenvectors requires a constant number of operations. Thus, each time a position is collected, the algorithm has $O(1)$ operations and can therefore be applied in real-time.

2.4.2 Extracting Linear Objects from Image Data

Imaging data can be produced from a vast array of modalities. Most commonly, ultrasound, computed tomography, and magnetic resonance imaging are used. In each imaging modality, different structures appear with different intensity values. Unfortunately, automatic image segmentation is a very difficult undertaking, and cannot be relied upon for accurate feature identification. Additionally, there is no natural ordering associated with voxels.

As a robust and convenient way to extract linear objects from images, a manual method is proposed. The user is required to place fiducial points on each linear object appearing in the image. Then, the linear object can be computed using the same principal components analysis

technique described above. Since there is no ordering associated with the imaging data or with the fiducial placement, the user is required to place fiducials on one linear object at a time and manually delineate the linear objects.

Figure 2.7: Images of axial (left), sagittal (middle), and coronal (right) of a reconstructed ultrasound volume of a Targeting Tutor phantom **[1]**, with fiducial points placed on one plane of the volume.

2.4.3 Extracting Linear Objects from Model Data

Often, when a well-defined structure is present in imaging data, it is segmented (either manually or automatically) from the rest of the image. It can then be converted into a geometrical model, which has the advantage that it is easier to visualize and is computationally more efficient to manipulate and store. Though models exhibit many similar properties to volumes, their surface properties can be calculated more readily, which lends to better linear object identification.

Surface models are made up of a set of surface points (called vertices), a set of finite lines connecting adjacent vertices, and a set of finite planes defined by the enclosing vertices and lines. To extract linear objects from models, a semi-automatic method is proposed. The desired dimensionality of the linear object must be specified, and then a fiducial point should be placed on the desired point, line, or plane. The nearest vertex, finite line, or finite plane (as appropriate) is found, and its equation is used as the equation for the point, line, or plane.

2.5 Iterative Convergence Algorithm

It can be shown (see Appendix A for proof) that the above proposed algorithm for registering points to points, lines to lines, and planes to planes always finds the globally optimal solution for a finite version of the problem. In practice, however, it is often the case that the linear objects in either or both of sets S_A and S_B are extracted from a set of points. This certainly is the case when the linear objects are extracted from tracking data, and also possibly the case when linear objects are extracted from image data. In this case, an iterative algorithm can be applied to determine the minimum sum of squares distances from the collected points to the linear objects.

To do this, an approach similar to the approach from Chen and Medioni [16] is used to iteratively register a point-set to a surface. In this case, the point-set $\vec{x}_A$ comes from the raw extracted data and the "surface" is the set of linear objects S_B, where the correspondence between each point in the point-set and a linear object $\varphi(\vec{x}_A)$ is known. Because correspondence is known and by linearity of the problem, each point in the point-set can be projected onto its corresponding linear object. Thus, the point of correspondence on the "surface" S_B to each point in the point-set $\vec{x}_A$ can be readily calculated by projection.

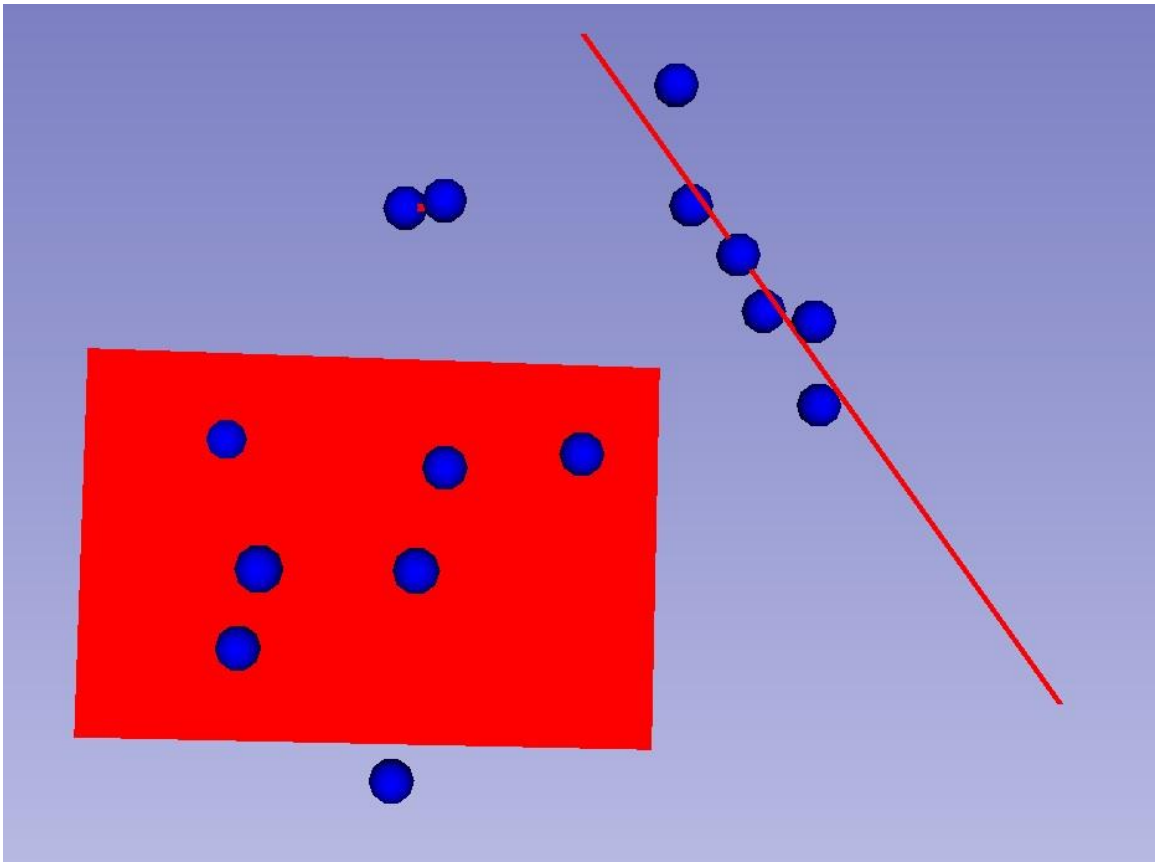

Figure 2.8: Illustration of iterative convergence step in proposed algorithm where collected points (blue) are registered to their corresponding point, line, or plane (red).

The algorithm used to converge to an optimal solution given a point-set with known correspondence to a set of linear objects is given below. It is assumed that point-to-point, line-to-line, and plane-to-plane registration has already been performed. This ensures that the initial guess is close to the globally optimal solution.

1. For each point $\vec{x}_{A,i}$ in the point-set $\vec{x}_A$, find the closest point $\vec{y}_{B,i}$ on its corresponding linear object $\hat{\varphi}(\vec{x}_{A,i})$ by projection.
2. Calculate the translational registration between coordinate frames by calculating the offset between the centroid $\vec{c}_A$ of the point-set $\vec{x}_A$ and the centroid $\vec{c}_B$ of the corresponding points $\vec{y}_B$.

3. For each point $\vec{x}_{A,i}$ in the point-set $\vec{x}_A$, find the closest point $\vec{y}_{B,i}$ on its corresponding linear object $\hat{\varphi}(\vec{x}_{A,i})$ by projection.
4. Calculate the spherical registration between the point-set $\vec{x}_A$ and the corresponding points $\vec{y}_B$.
5. Iterate until the change in the fiducial registration error is below some threshold ε.

This iterative algorithm is guaranteed to converge to at least a local minimum (this follows directly from the proof in Besl and McKay [15]), and likely will converge to a global minimum due to its initial starting point of iteration. This algorithm is described above for the case when one set of linear objects has been extracted from a point-set, but can also be applied when both sets of linear objects have been extracted from a point-set. This is done by using in the calculations in steps 2 and 4 the point-sets $\vec{z}_A = \vec{x}_A \cup \vec{y}_A$, where $\vec{x}_A$ is the initial point-set in A and $\vec{y}_A$ is the points corresponding to $\vec{x}_B$, and $\vec{z}_B = \vec{x}_B \cup \vec{y}_B$, where $\vec{x}_B$ is the initial point-set B and $\vec{y}_B$ is the points corresponding to $\vec{x}_A$.

2.6 Summary of Linear Object Registration Algorithm

The following workflow summarizes the linear object registration algorithm:

1. Extract linear objects from raw medical imaging data.
2. Select a set of references in each coordinate frame.
3. Find linear object correspondence by matching the vectors of distances to the set of references (i.e. signatures).
4. Calculate the linear object centroid in each coordinate frame.
5. Perform point-set registration using the centroid projections and the direction/normal vectors (matched using the set of references).
6. If either or both sets of linear objects is extracted from a point-set, use the iterative convergence algorithm to find the least-squares registration.

Step 1 is the linear object extraction phase, steps 2-5 are the point-to-point, line-to-line, and plane-to-plane registration phase, and step 6 is the iterative convergence phase.

Two important observations about this algorithm are: (1) Each step is polynomial complexity, and thus, the entire algorithm is polynomial complexity, (2) Each step is guaranteed to produce a unique result given a problem that admits a unique registration, and thus, the entire algorithm is guaranteed to produce a unique solution. Further, the algorithm is guaranteed to converge to a globally optimal solution to a finite version of the registration problem, and thus, it approaches the globally optimal solution to the infinite problem as the noise amplitude approaches zero.

Chapter 3

IMPLEMENTATION

3.1 Open-Source 3D Slicer Implementation

To make the proposed linear object registration algorithm a usable tool for medical computing and computer-assisted interventions, the linear object registration algorithm was implemented as a practical registration tool in the SlicerIGT extension (www.slicerigt.org) for 3D Slicer (www.slicer.org), a free and open-source medical image visualization and computing platform. This way, the implementation builds upon the visualization and registration features already available in 3D Slicer. The SlicerIGT extension is a collection of modules related to image-guided therapy for navigation in minimally-invasive medical interventions. By implementing the linear object registration algorithm as a SlicerIGT module for 3D Slicer, this makes the algorithm widely available to the medical image computing community and allows it to be used without restriction. Additionally, it allows the module to use many of the imaging and tracking features and interfaces already integrated in the SlicerIGT extension. For many practical image-guided applications, Slicer interfaces with the PLUS software library for ultrasound-guided tracked interventions. PLUS is used to interface with tracking and imaging devices, and 3D Slicer interfaces with PLUS to receive processed data.

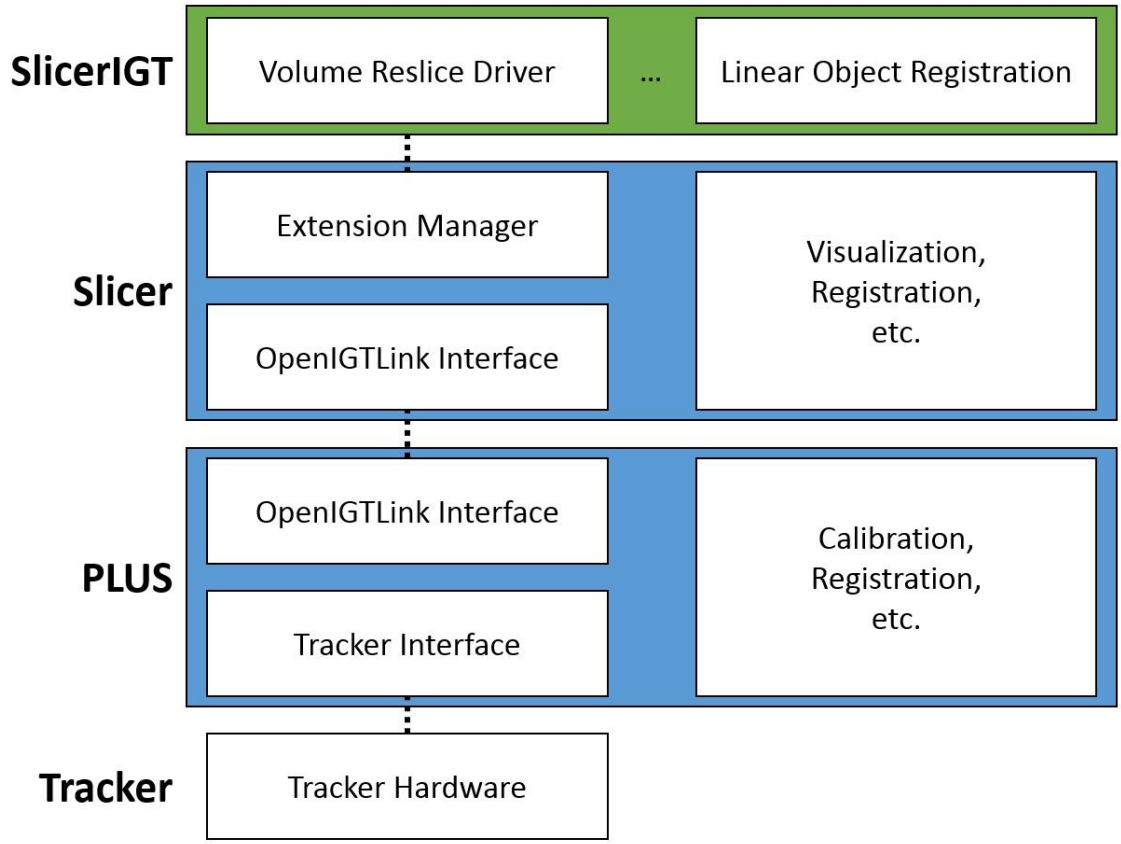

Figure 3.1: PLUS, Slicer, and SlicerIGT architecture used for image-guided therapy applications. The linear object registration module is designed to integrate with this architecture.

The SlicerIGT module for linear object registration is designed to guide users through the linear object registration process by allowing users to visualize the process. This includes showing linear object correspondence, convenient user interfaces for extracting linear objects from medical data, visualizing the linear objects, and visualizing the registration result. Additionally, the user may optionally adjust the algorithm parameters.

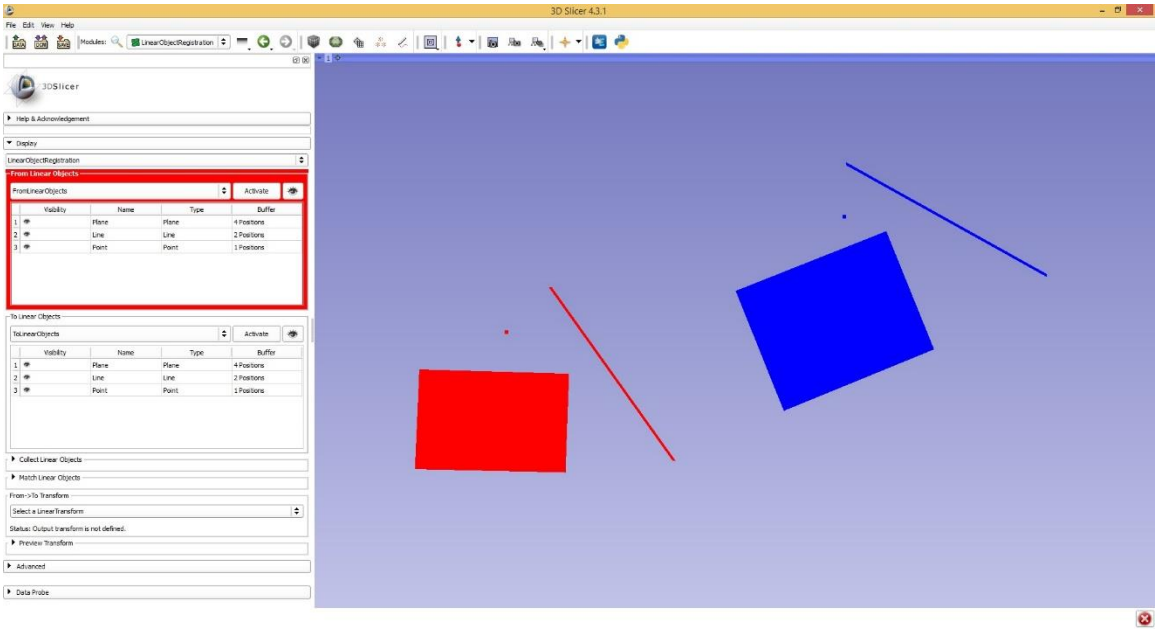

Figure 3.2: Screenshot of linear object registration module for 3D Slicer.

3.2 Linear Object Collection

As described in Section 2.4, there are several ways in which a user may collect linear objects. In addition to providing an interface for the linear object registration algorithm, one of the key goals of the module is to provide an interface for collecting linear objects. To this end, the module offers five collection modes: 1) manual degrees of freedom, 2) manual segmentation, 3) automatic segmentation, 4) model, 5) fiducial.

3.2.1 Manual Degrees of Freedom Collection Mode

This collection mode is intended for collecting linear object using a tracked stylus. The user must manually indicate for each linear object when collection starts and when collection stops. The user must additionally indicate what type of linear object is being collected (the linear object is automatically resolved by the algorithm outlined in Subsection 2.4.1).

Figure 3.3: Interface for collecting manually marked linear objects with a tracked stylus.

3.2.2 Manual Segmentation Collection Mode

This collection mode is intended for collecting linear objects using a tracked stylus. The user must manually indicate for each linear object when collection starts and stops, but need not manually indicate what type of linear object is being collected. The type of linear object collected is automatically resolved by the algorithm described in Subsection 2.4.1.

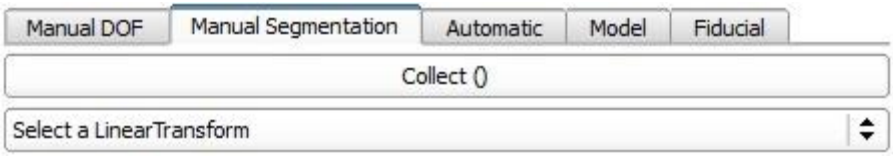

Figure 3.4: Interface for collecting manually delineated linear objects with a tracked stylus.

3.2.3 Automatic Segmentation Collection Mode

This collection mode is intended for collecting linear objects using a tracked stylus. The algorithm automatically determines when the user is collecting a linear object and what type of linear object the user is collecting. This is accomplished using the algorithm described in Subsection 2.4.1.

Figure 3.5: Interface for collecting automatically identified linear objects using a tracked stylus.

3.2.4 Model Collection Mode

This collection mode is intended for selecting known linear objects on a model in the 3D Slicer scene. The user need only select the type of linear object they wish to collect and click the desired linear object on the model. The module automatically computes the closest linear object of the specified type to where the user clicked. This is accomplished using the algorithm described in Subsection 2.4.3.

Figure 3.6: Interface for collecting linear objects from a geometrical model.

3.2.5 Fiducial Collection Mode

This collection mode is primarily intended for collecting linear object in volumes, but may also be applicable for several other applications (for example, collection with a tracked stylus). The user may place a set of fiducials on the linear object in the volume they wish to collect. Then, the module automatically fits a linear object of the appropriate type to the set of fiducials (using the algorithm described in Subsection 2.4.2).

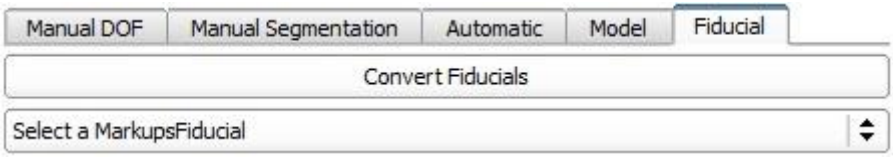

Figure 3.7: Interface for collecting linear objects from a volume by placing fiducials on each linear object appearing in the volume.

Chapter 4

VALIDATION

4.1 Introduction to Validation

To validate the accuracy and usefulness of the proposed algorithm, the algorithm must be tested in several scenarios, including scenarios that cover each of the data types from which linear objects can be extracted. This means the algorithm must be validated for registration of tracking data, image data, and model data.

Importantly, to validate the algorithm's accuracy, the registration result the algorithm produces must be compared to a ground-truth result. In practical registration, no such ground-truth is available. To circumvent this problem, several methods were used. First, simulated data was generated for which the ground-truth is well known. Second, the linear object registration was compared to point-set registration using well-defined fiducial points. Finally, several registration quality metrics were used to evaluate the registration.

To compare a calculated registration to a ground-truth registration, measures of matrix similarity were used. Since the registration is a 4-by-4 homogenous transformation matrix, the translational and rotational components can each be readily extracted. The norm of the difference in translational components is taken as a measure of translational similarity. The angle of the quotient of the rotational components (computed by converting to axis angle form) is taken as a measure of rotational similarity. Supposing a transformation matrix T can be decomposed into a rotational part R and a translational part $\vec{d}$, these metrics are expressed below.

$$sim_{TRANS}(T_1, T_2) = dist(\vec{d}_1, \vec{d}_2)$$

$$sim_{ROT}(T_1, T_2) = arccos\left(\frac{tr({R_1}^T R_2) - 1}{2}\right)$$

As a measure of registration quality when a ground-truth registration was unavailable, the target registration error was used.

Here, four validation experiments were performed: 1) simulated data for registration of a tracked phantom to a navigation coordinate frame, 2) registration of a tracked phantom to a model or volume in a navigation coordinate frame, 3) registration of a tracked phantom to a model in a navigation coordinate frame when no fiducial points are available, 4) registration of a reconstructed ultrasound volume to a model. The methods and results for each experiment are described subsequently.

4.2 Validation with Simulated Data

As an initial form of validation, simulated data was generated. This data has the advantage that it can be manufactured with known ground-truth, but may not comprehensively model the noise associated with real data. The simulated data was generated in the context of phantom registration, where it was assumed that one set of linear objects was defined by the phantom's geometry in coordinate frame B and the other set of linear objects was collected by a tracked stylus in coordinate frame A.

4.2.1 Generating Simulated Data

The objective of the simulated data generator was to create a set of linear objects with random position and orientation, then randomly transform them, and sample points with added noise from the transformed linear objects. To this end, the simulator proceeded as follows, assuming the desired number of points, lines, planes, and references were specified by N_P, N_L, N_A, and N_R and the root-mean-square noise is specified by ε_{SIM}:

1. For each of N_P points, N_L lines, N_A planes, and N_R references, one, two, three, and one random position(s) respectively were generated in space generated from a uniform

distribution within the range of a typical phantom. Each linear object was defined such that each generated position was on the linear object. These were the linear objects defined in coordinate frame B.

2. Each linear object was parameterized. For each linear object, parameters were randomly generated from a uniform distribution, and the point defined by those parameters was added to the point-set $\vec{x}_B$.
3. A random rotation R was generated by creating a random matrix M with each element on the uniform distribution $[0,1]$. The singular value decomposition of M was computed as $M = UDV^T$, where U and V were necessarily singular. Then, the random rotation was calculated as $R = UV^T$, where R is rejected if $det(R) \neq +1$.
4. A random translation $\vec{d}$ was generated by sampling each component from a uniform distribution within the range of a typical phantom.
5. Each element of the point-set $\vec{x}_B$ was transformed into coordinate frame A, by $\vec{x}_A = R\vec{x}_B + \vec{d} + \vec{n}$, where $\vec{n}$ was a noise vector with components taken from a Gaussian distribution with amplitude ε_{SIM}.

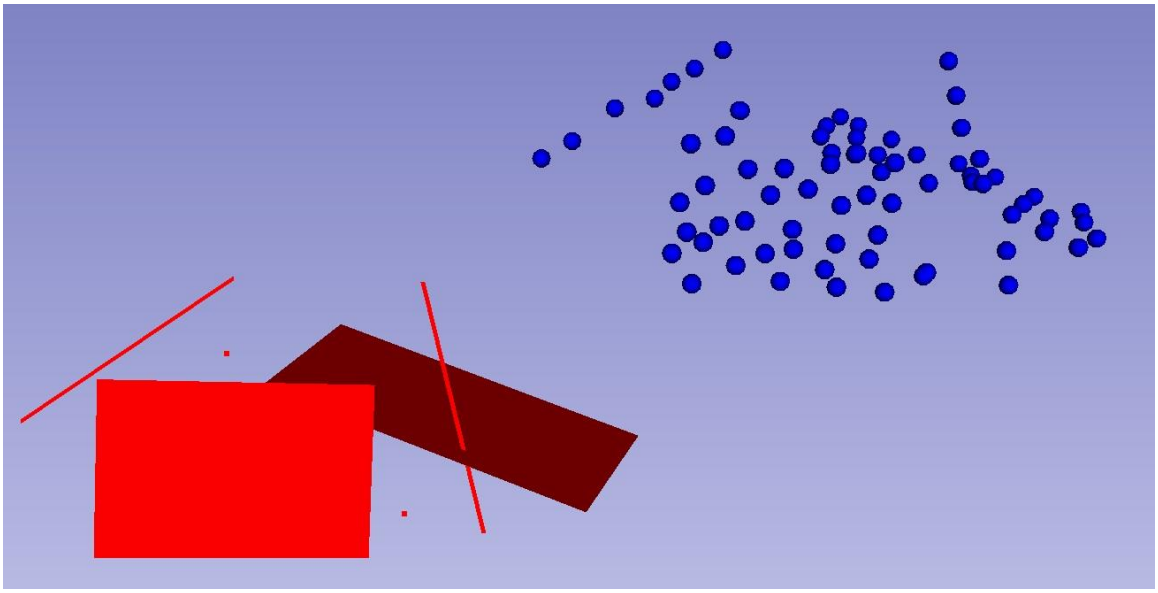

Figure 4.1: Example illustration of simulated data, with linear objects defined in coordinate frame ***B*** (red) and collected points defined in coordinate frame ***A*** (blue).

This workflow generated a set of linear objects "defined" in coordinate frame B (simulating the linear objects defined by the phantom's geometry) and a set of points collected on linear objects in coordinate frame A (simulating linear objects collected by a tracked stylus). Information about which subsets of $\vec{x}_A$ correspond to a single linear object was known, but the exact correspondences remained unknown to the registration algorithm. Importantly, the ground-truth transformation defined by R and $\vec{d}$ was known. Thus, the registration results were compared to it.

4.2.2 Validation with Simulated Data: Results

First, the number of linear objects was fixed at $N_P = N_L = N_A = N_R = 4$, and the noise ε_{SIM} was varied. For each level of noise, the translational and rotational similarity metrics were calculated over fifty repetitions. The results are displayed in Figure 4.2.

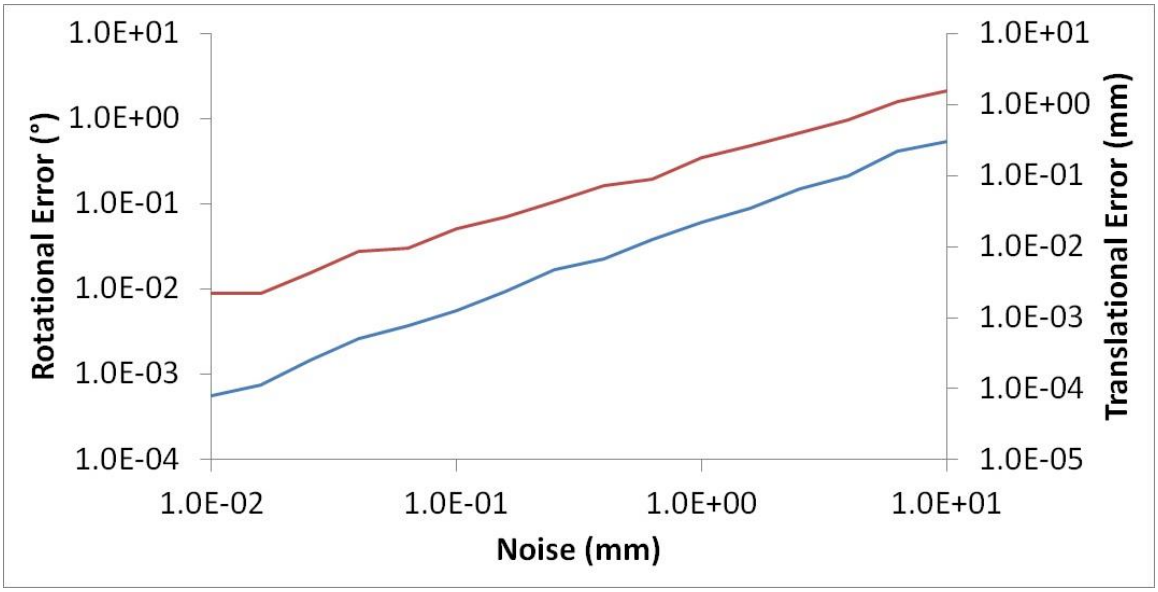

Figure 4.2: Log-log plot of similarity metrics with respect to root-mean-square noise for the simulated data. Rotational error is indicated in blue and translational error is indicated in red.

The most important observation from this data is that the similarity metrics increase linearly with the noise associated with the points in set $\vec{x}_A$. This demonstrates the robustness and stability of the algorithm with respect to noise. Since it has been shown that in the case of no noise the proposed algorithm achieves the globally optimal solution (see Appendix A for proof), these results provide empirical evidence that the registration error scales linearly with noise. The similarity metrics for particular levels of noise corresponding to tool tracking technologies should be noted. Table 4.1 reports the mean translational and rotational similarity metrics for the simulated data for levels of noise associated with common commercial tool tracking systems.

Table 4.1: Rotational and translational error in linear object registration of simulated data for noise amplitudes associated with various commercial tool tracking systems.

Tracker	Reported RMS Error (mm)	Rotational Error (°)	Translational Error (mm)
Claron MicronTracker	0.20	0.01	0.03
NDI Polaris Spectra	0.25	0.02	0.03
NDI Aurora	0.70	0.04	0.08
Ascension TrakStar	1.40	0.07	0.16

To determine the number of references required to sufficiently define the matching between linear objects in two coordinate frames, the number of references N_R was varied, while the root-mean-square noise ε_{SIM} was held fixed at 1.4mm (the root-mean-square noise associated with the least accurate tracking system). For all simulated data experiments where $N_R \geq 1$, the references were sufficient to correctly determine correspondence between the linear objects. This indicates that using only one reference may be sufficient for determining linear object correspondence in the majority of practical scenarios, though using more references is strongly recommended.

4.3 Validation with Tool Tracking Data

Recently, there has been a shift in the medical training curriculum is to focus on anatomical simulation and phantom-based training. Augmenting this training with a 3D navigation view has been shown to improve the training process [1]. To visualize the surgical tools in the same coordinate system as the phantom, phantom registration must be performed. This also allows objective metrics to be calculated for the procedure [1].

Figure 4.3: Perk Tutor system configured for use in lumbar puncture training experiments. The phantom, needle, and ultrasound probe are all registered to a common navigation space.

For these applications, registration is performed by identifying features on the phantom using a tracked stylus, and identifying the same features on the phantom in the navigation coordinate system by mouse-click. Typically, this is achieved using point features and point-set registration. To validate the proposed linear object registration algorithm, it was used for registration and compared to the point-set registration which was used as the "gold standard".

4.3.1 Phantom to Model Registration with Tracked Tools: Methods

To localize features of a phantom with a tracked stylus, first a calibration was performed to determine the tip of the stylus. The transformation from the sensor on the stylus to the stylus

tip was calculated automatically by pivoting the stylus about its tip [32]. Then, the stylus tip was used to localize features.

The phantoms used here were an fCal 1.2 ultrasound calibration phantom [33] and a lumbar spine phantom [1]. Both of these phantoms have landmark points manufactured on their exterior intended for point-set registration purposes. Additionally, each phantom has well-defined lines and/or planes which too could be used for registration.

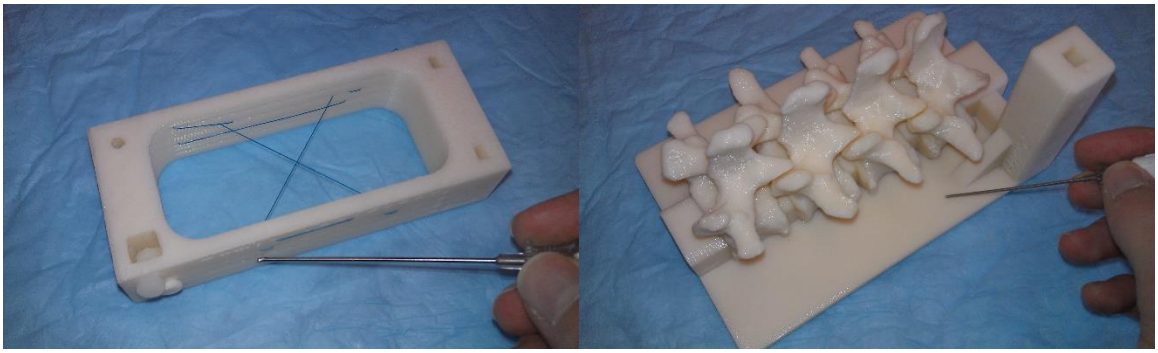

Figure 4.4: User collecting points using a tracked stylus on the fCal ultrasound calibration phantom (left) and the lumbar spine phantom (right).

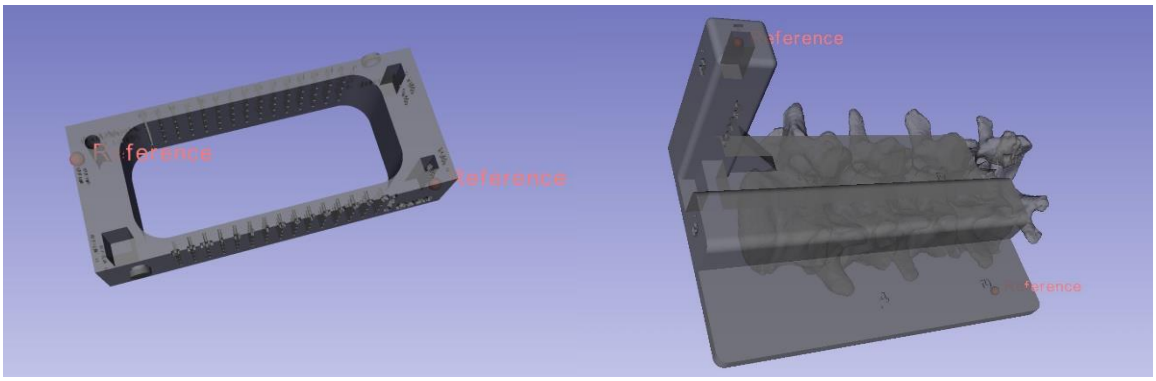

Figure 4.5: Reference points used to determine linear object correspondence for the fCal ultrasound calibration phantom (left) and the lumbar spine phantom (right).

The fCal 1.2 ultrasound calibration phantom has a box-like exterior, including six exterior faces. Each face of the exterior was used as a plane for linear object registration. A subset of the landmark points were used as references to find the linear object correspondences. The lumbar spine phantom has a spine which rests on a rectangular base. Each face of the rectangular base was used as a plane for linear object registration. Additionally, there is a mount which

attaches the lumbar vertebrae to the base. The intersection of the base and the mount forms two lines which were used for linear object registration. A subset of the landmark points were used as references to find the linear object correspondences.

To validate the linear object registration algorithm, the landmark points were first used to perform point-set registration. This registration result was used as the "gold standard". Then, the linear object registration was performed and compared to the "gold standard" using the transformation similarity metrics.

All experiments were performed using the Ascension DriveBay electromagnetic tracker (www.ascension-tech.com), which reports position values with 1.4mm root-mean-square accuracy and rotation values with 0.5° root-mean-square accuracy.

4.3.2 Phantom to Model Registration with Tracked Tools: Results

For both phantoms, the point-set registration was repeated twelve times for the fCal ultrasound calibration phantom and nine times for the lumbar spine phantom, and the "mean" transformation (calculated using average rotation quaternion) was used as the "gold standard". Then, the linear object registration was repeated, for each phantom, an equal number of times and the average translational and rotational similarity metrics were computed for both phantoms (Table 4.2).

Table 4.2: Rotational and translational error metrics compared to the ground-truth transformation calculated using point-set registration for the fCal ultrasound calibration phantom and the lumbar spine phantom.

Surgical Tool	Rotational Error (°)	Translational Error (mm)
fCal Phantom	1.49	0.74
Lumbar Spine Phantom	0.76	1.15

Table 4.3: Rotational and translational precisions for each registration algorithm for the fCal ultrasound calibration phantom. The precision is calculated as the mean difference between each registration and the mean registration.

Registration Method	Rotational Precision (°)	Translational Precision (mm)
Point-Set Registration	0.46	0.45
Linear Object Registration	0.43	0.37

Table 4.4: Rotational and translational precisions for each registration algorithm for the lumbar spine phantom. The precision is calculated as the mean difference between each registration and the mean registration.

Registration Method	Rotational Precision (°)	Translational Precision (mm)
Point-Set Registration	0.29	0.35
Linear Object Registration	0.42	0.76

Importantly, the translational similarity metrics were less than the tracking error. Thus, this shows that the difference in the translational components of the registration is trivial. The rotational similarity metrics, on the other hand, were not smaller than the tracking error, but still represent a practically insignificant difference in registration results.

Interestingly, the root-mean-square registration error was lower for linear object registration for both phantoms (1.22mm vs. 2.13mm for the fCal 1.2 ultrasound calibration phantom and 1.14mm vs. 1.33mm for the lumbar spine phantom).

For the Matlab implementation, the algorithm took 34s on average for the fCal phantom (9,816 collected points on average) and 71s on average for the lumbar spine phantom (11,969 collected points on average).

4.3.3 Phantom to Volume Registration with Tracked Tools: Methods

Similar to phantom registration experiments, volumes can be registered using a tracked stylus. Instead of the above phantom registration experiments where the phantoms are 3D printed

from computerized models, arbitrary surgical tools with linear features can be imaged and their imaging volume can be registered to the navigation space.

For this experiment, a surgical targeting tool was registered to the navigation space [34]. This targeting tool was designed to measure needle placement accuracy of image-guided needle placements, consisting of a series of target points embedded in a block of opaque gel. The targeting tool was imaged by CT scan. Then, the CT scan was segmented, and the segmented CT scan was registered to the navigation space.

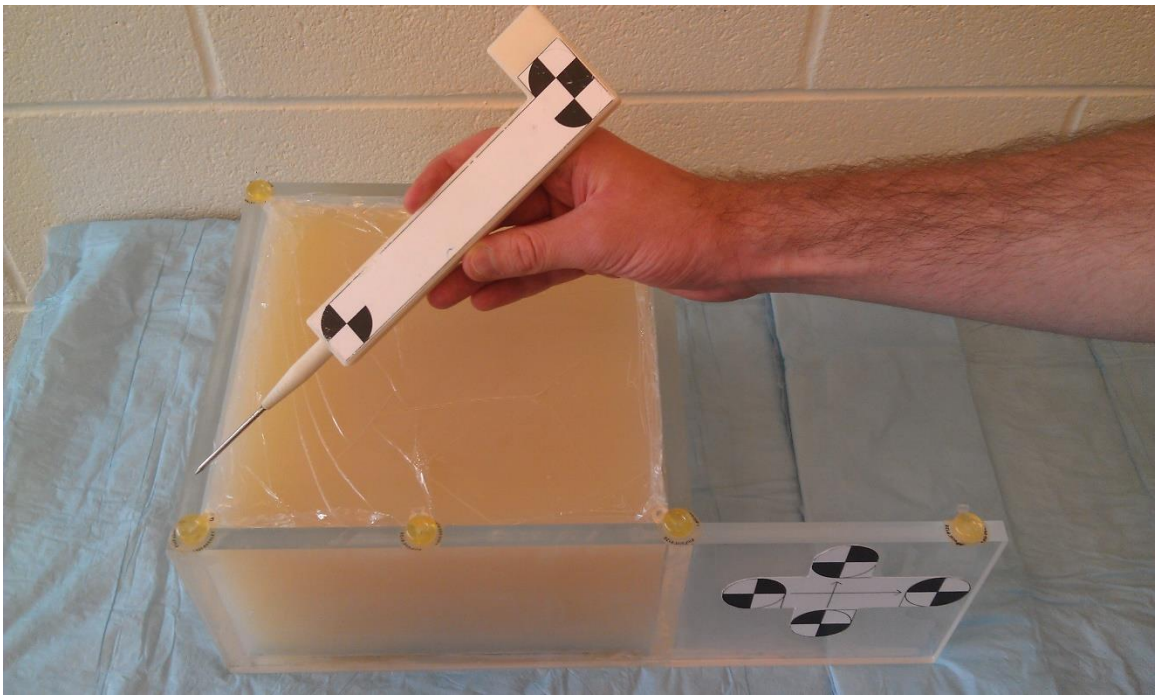

Figure 4.6: User collecting points using a tracked stylus on the gel block targeting tool.

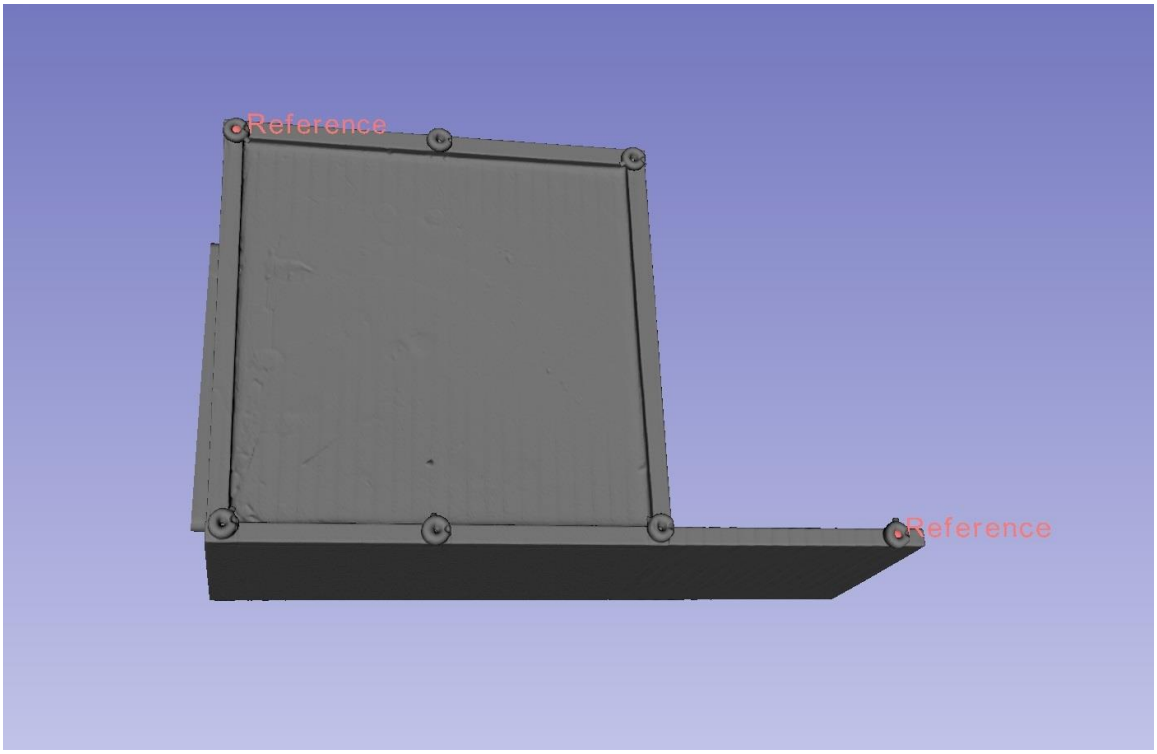

Figure 4.7: Reference points used to determine linear object correspondence for the gel block targeting tool.

The gel block targeting tool has a cuboid exterior, including six exterior faces. Each face of the exterior (except the inferior face) was used as a plane for linear object registration. A subset of the landmark points were used as references to find the linear object correspondences. Again, validation was performed by comparing the "gold standard" point-set registration (with the landmark points placed on the phantom for registration purposes) to the linear object registration using the transformation similarity metrics.

All experiments were performed using the Claron MicronTracker H360 optical tracker (www.clarontech.com), which reports position values with 0.20mm root-mean-square accuracy.

4.3.4 Phantom to Volume Registration with Tracked Tools: Results

Again, the point-set registration was repeated nine times, and the "mean" transformation (calculated using average rotation quaternion) was used as the "gold standard". Then, the linear

object registration was repeated an equal number of times and the average translational and rotational similarity metrics were computed for the targeting tool (Table 4.5).

Table 4.5: Rotational and translational error metrics compared to the ground-truth transformation calculated using point-set registration for the gel block targeting phantom.

Surgical Tool	Rotational Error (°)	Translational Error (mm)
Gel Block Phantom	0.27	0.74

Table 4.6: Rotational and translational precisions for each registration algorithm for the gel block targeting phantom. The precision is calculated as the mean difference between each registration and the mean registration.

Registration Method	Rotational Precision (°)	Translational Precision (mm)
Point-Set Registration	0.08	0.21
Linear Object Registration	0.09	0.24

Interestingly, the translational and rotational similarity metrics were slightly larger than the tracking error. Though, it is noted that they were less than the fiducial registration error (0.92mm for point-set registration) associated with the point-set registration. The root-mean-square registration error was, once again, lower for linear object registration (0.52mm vs. 0.92mm).

For the Matlab implementation, the algorithm took 3.3s on average for registration of the gel block targeting tool (1438 collected points on average).

4.4 Validation with Ultrasound Calibration Data

Ultrasound calibration is integral for tracked image-guided interventions. This allows the operator to visualize the ultrasound probe and image, along with the tracked needle in a common navigation space. Using this form of navigation has been shown to improve operator performance over traditional ultrasound techniques [35]. Ultrasound calibration is a three step procedure [32]:

1. Perform pivot calibration to determine the stylus tip location.
2. Perform phantom registration to determine the phantom's position in navigation space using the tracked stylus (typically achieved by point-set registration).
3. Perform N-wire calibration to determine the position of the image relative to the phantom.

4.4.1 Phantom to Model Registration for Ultrasound Calibration: Methods

The phantom used here was a LEGO® brick ultrasound calibration phantom [6]. This phantom does not have landmark points explicitly manufactured on its exterior. Several landmark points on the phantom's exterior may be estimated, but this may be inaccurate. The phantom does, however, have well-defined lines and/or planes which could be used for registration.

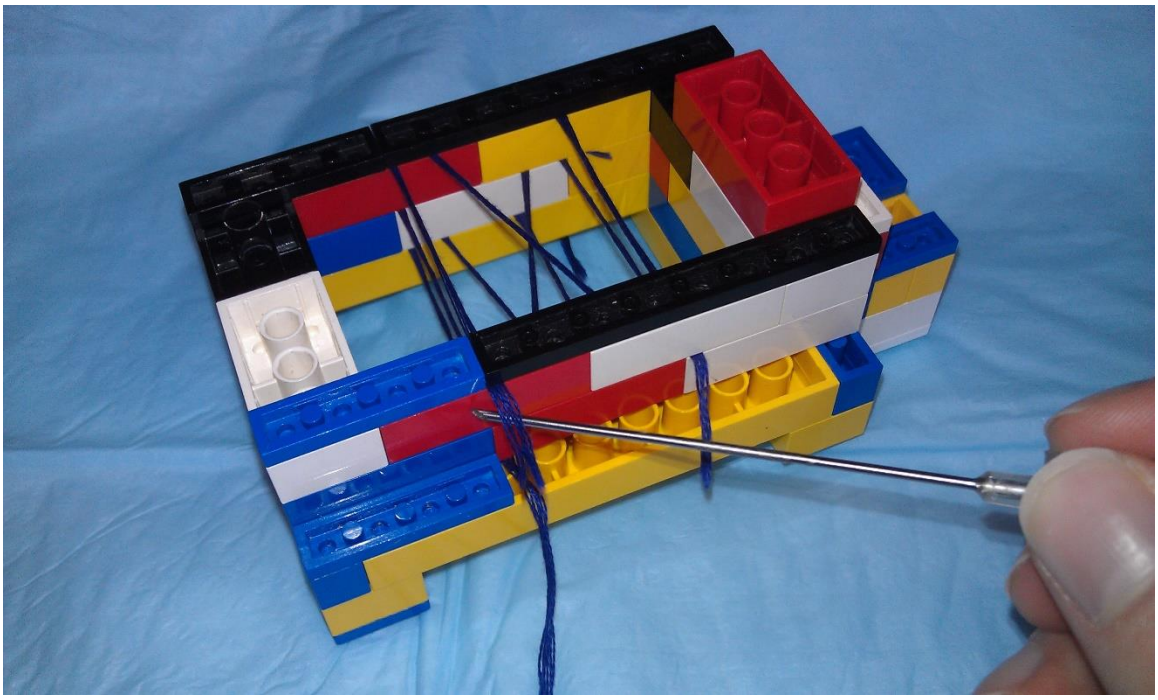

Figure 4.8: User collecting points using a tracked stylus on the LEGO® brick ultrasound calibration phantom.

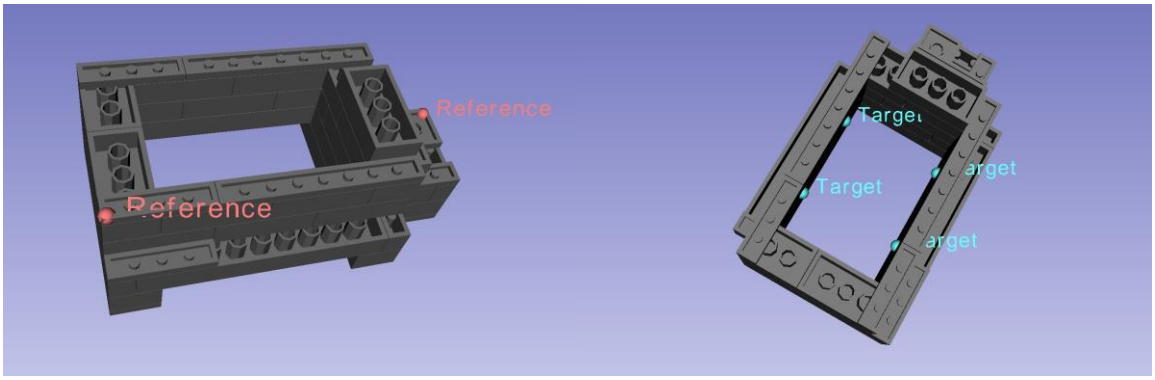

Figure 4.9: Reference points used to determine linear object correspondence (left) and target points used to evaluate target registration error (right) for the LEGO® brick ultrasound calibration phantom.

The LEGO® brick ultrasound calibration phantom has a box-like exterior, including six exterior faces. Each face of the exterior was used as a plane for linear object registration. A subset of the landmark points were used as references to find the linear object correspondences.

Point-set registration was performed using the estimated landmark locations identified in previous studies [6] involving the LEGO® brick ultrasound calibration phantom. In this case, the landmarks could not be collected accurately, so the point-set registration could not be used as a "gold standard". Instead, two metrics of registration quality were used.

First, the target registration error [8] was used to quantify the registration error associated with target points of anatomy. The location where the four corner wires intersect the phantom walls were used as target points.

Additionally, since the chosen phantom is an ultrasound calibration phantom, the point reconstruction accuracy (PRA), a measure of the distance from a target point's ground-truth position to its reconstructed position based on its location in the ultrasound plane [36], was also used. To compute the point reconstruction accuracy, a cross-wire was imaged with the tracked ultrasound and segmented to find the point $\vec{x}_U$ in the ultrasound coordinate frame U and probed with the tracked stylus to find the point $\vec{x}_R$ in the reference coordinate frame R. The point reconstruction accuracy was calculated (where $T_{U\to P}$, the transformation from the ultrasound

coordinate frame to the probe coordinate frame, was calculated by the calibration and $T_{P\to R}$, the transform from the probe coordinate frame to the reference coordinate frame, was given by the tracking system).

$$PRA = |\vec{x}_R - T_{P\to R}T_{U\to P}\vec{x}_U|$$

The two different metrics were compared assuming that the pivot calibration and image calibration sequences are identical for all phantom registrations.

All experiments were performed using the Ascension DriveBay electromagnetic tracker (www.ascension-tech.com), which reports position values with 1.4mm root-mean-square accuracy and rotation values with 0.5° root-mean-square accuracy.

4.4.2 Phantom to Model Registration for Ultrasound Calibration: Results

The phantom registration was performed using both registration methods (point-set and linear object) twelve times each. The target registration error was computed as the average target registration error over all targets and all registrations. The point reconstruction accuracy was computed as the average over all registrations, and the cross-wire phantom was imaged and probed twenty times each. To reduce extraneous sources of error, the point reconstruction accuracy was computed using the same sequence of tracked ultrasound frames each time, but changing the phantom registration.

Table 4.7: Target registration error and point reconstruction accuracy for ultrasound calibrations with the LEGO® brick ultrasound calibration phantom using both point-set registration and linear object registration.

Registration Method	Target Registration Error (Mean ± SD mm)	Point Reconstruction Accuracy (Mean ± SD mm)
Point-Set Registration	1.34 ± 0.31	3.98 ± 0.64
Linear Object Registration	1.18 ± 0.22	3.46 ± 0.80

Table 4.8: Rotational and translational precisions for each registration algorithm for the LEGO® brick ultrasound calibration phantom. The precision is calculated as the mean difference between each registration and the mean registration.

Registration Method	Rotational Precision (°)	Translational Precision (mm)
Point-Set Registration	0.22	0.66
Linear Object Registration	0.25	0.32

For both error metrics, the linear object registration produced more accurate results (both significant by Student's *t*-test and verified by Mann-Whitney U test, both with $\alpha = 0.05$). This leads to the conclusion that linear object registration is more accurate when the landmark points cannot be accurately identified.

Again, the root-mean-square registration error was lower for linear object registration (0.45mm vs. 0.53mm).

For the Matlab implementation, the algorithm took 57s on average for registration of the LEGO® brick ultrasound calibration phantom (10,380 collected points on average).

4.5 Validation with Reconstructed Volume Registration

Once ultrasound calibration has been performed, it is possible to use the ultrasound images, along with tool tracking information to reconstruct a volume from the tracked images. Volume reconstruction can be useful for allowing operators to visualize anatomy in 3D, using only ultrasound. Unfortunately, ultrasound volume reconstruction can be inaccurate due to ultrasound image calibration error and tracking error. For the validation experiments described here, the ultrasound image calibration method proposed by Welch *et al.* [37] was performed, and the volume reconstruction was done using the algorithm proposed by Vaughan *et al.* [38], which is implemented in the PLUS library (www.plustoolkit.org).

To validate the linear object registration algorithm, the volume reconstruction of a phantom by tracked ultrasound can be registered to a model of the phantom with known geometry. This potentially has applications in phantom registration where the phantom is already embedded in a gel. Due to error propagation in the volume reconstruction, however, this registration could be highly inaccurate, and thus, the results here should only be considered as a proof-of-concept for volume registration using the proposed linear object registration algorithm.

4.5.1 Reconstructed Volume Registration: Methods

For this validation, the Targeting Tutor phantom, originally developed by Ungi *et al.* [1], was used. The phantom has four sets of four posts, two sets of which have flat tops and two sets of which have circular tops. Each set of flat-topped points was used to define a plane, and the apex of each circularly-topped post was used to define a point. Additionally, the top of the base of the phantom defined a plane and the two interior sides of the sensor post defined planes. Linear objects were manually segmented in 3D Slicer using 3D volume renderings of the reconstructed ultrasound volumes.

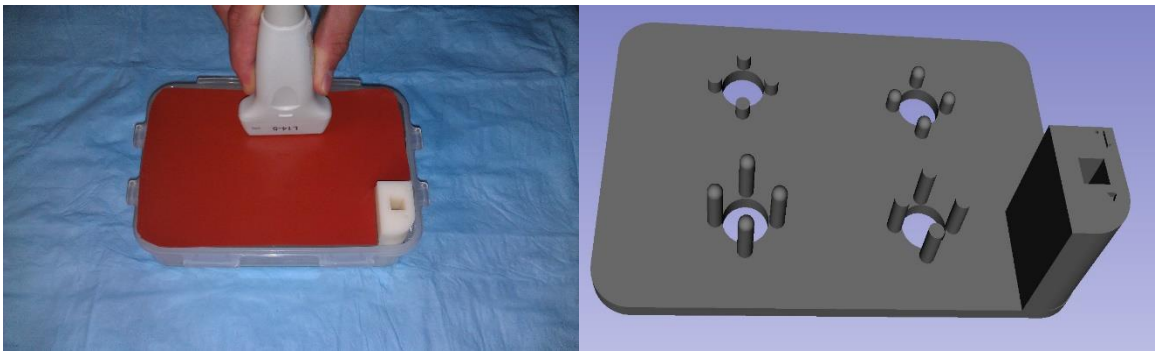

Figure 4.10: Photograph of user scanning the Targeting Tutor phantom using a tracked ultrasound probe (left) and image of Targeting Tutor phantom model (right).

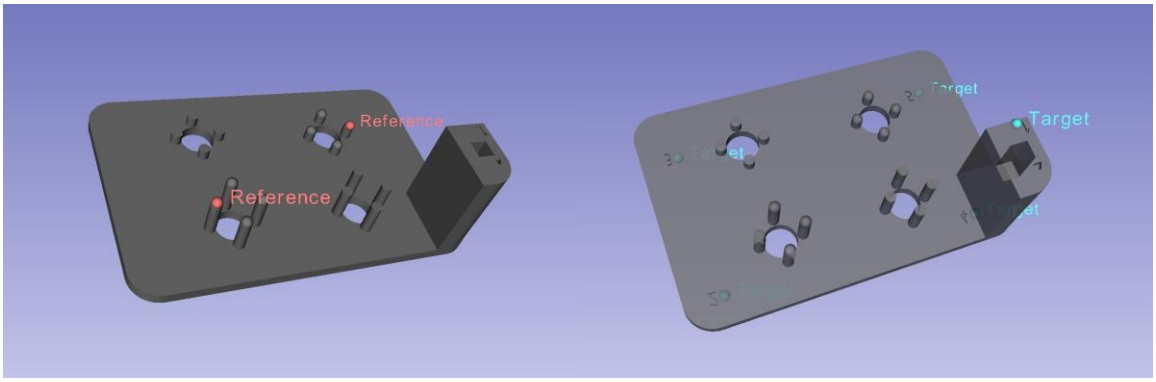

Figure 4.11: Reference points used to determine linear object correspondence (left) and target points used to evaluate target registration error (right) for the Targeting Tutor phantom.

First, point-set registration was used to register the reconstructed volume to the phantom model with known geometry. The eight points defined by the apices of the circularly-topped posts were used as fiducial points for the point-set registration. Subsequently, all of the other defined linear objects (the base plane, the two interior planes of the sensor post, and the two top planes of the flat-topped posts) were used in conjunction with the eight apex points in the linear object registration algorithm.

Because the ultrasound volume was reconstructed in the coordinate system defined by the reference sensor mounted on the phantom, it was possible to also use this sensor to choose target points on the phantom. To this end, five target points were selected in the reference sensor coordinate system using a tracked stylus. These target points were known with significantly greater accuracy than the volume reconstruction and were used to compute the target registration error associated with both the point-set volume registration and the linear object volume registration.

All experiments were performed using the Ascension DriveBay electromagnetic tracker (www.ascension-tech.com), which reports position values with 1.4mm root-mean-square accuracy and rotation values with 0.5° root-mean-square accuracy. The Ultrasonix SonixTablet ultrasound machine (www.ultrasonix.com) with L14-5 linear probe was used to collect tracked ultrasound images.

4.5.2 Reconstructed Volume Registration: Results

In total, nine tracked ultrasound image sequences were collected and used to reconstruct ultrasound volumes of the Targeting Tutor phantom.

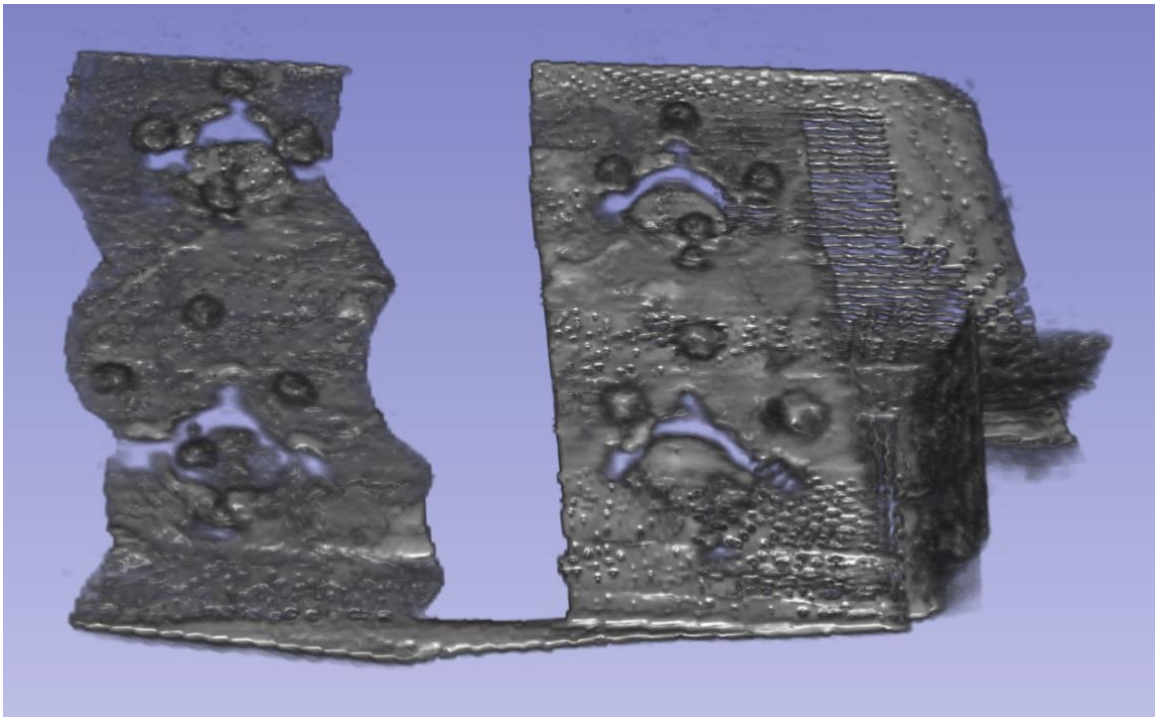

Figure 4.12: Example volume reconstruction of the Targeting Tutor phantom using tracked ultrasound.

For each reconstructed volume, point-set registration was performed and subsequently linear object registration was performed. The overall target registration error was calculated as the average over all targets, over all trials (Table 4.9). Targets were each probed ten times with a tracked stylus to ensure their position was known accurately.

Table 4.9: Target registration error of reconstructed ultrasound volume to model of the Targeting Tutor phantom using both point-set registration and linear object registration.

Registration Method	Target Registration Error (Mean ± SD mm)
Point-Set Registration	4.05 ± 1.57

Linear Object Registration	3.87 ± 0.91

Table 4.10: Rotational and translational precisions for each registration algorithm for the Targeting Tutor phantom reconstructed volume registration. The precision is calculated as the mean difference between each registration and the mean registration.

Registration Method	Rotational Precision (°)	Translational Precision (mm)
Point-Set Registration	1.01	1.63
Linear Object Registration	0.60	1.49

It is observed that the mean target registration error for both registration algorithms was approximately the same (effect size 0.14 by Cohen's d statistic), and shows no statistically significant difference (by Student's *t*-test and verified by Mann-Whitney U test, both with $\alpha = 0.05$). What is notable is that the standard deviation was significantly smaller for linear object registration (significant by F-test, with $\alpha = 0.05$). This indicates that the linear object registration algorithm offers better precision. This can be confirmed by looking at a histogram of target registration errors (Figure 4.13), which shows that outliers with large target registration error occurred less frequently with linear object registration than with point-set registration.

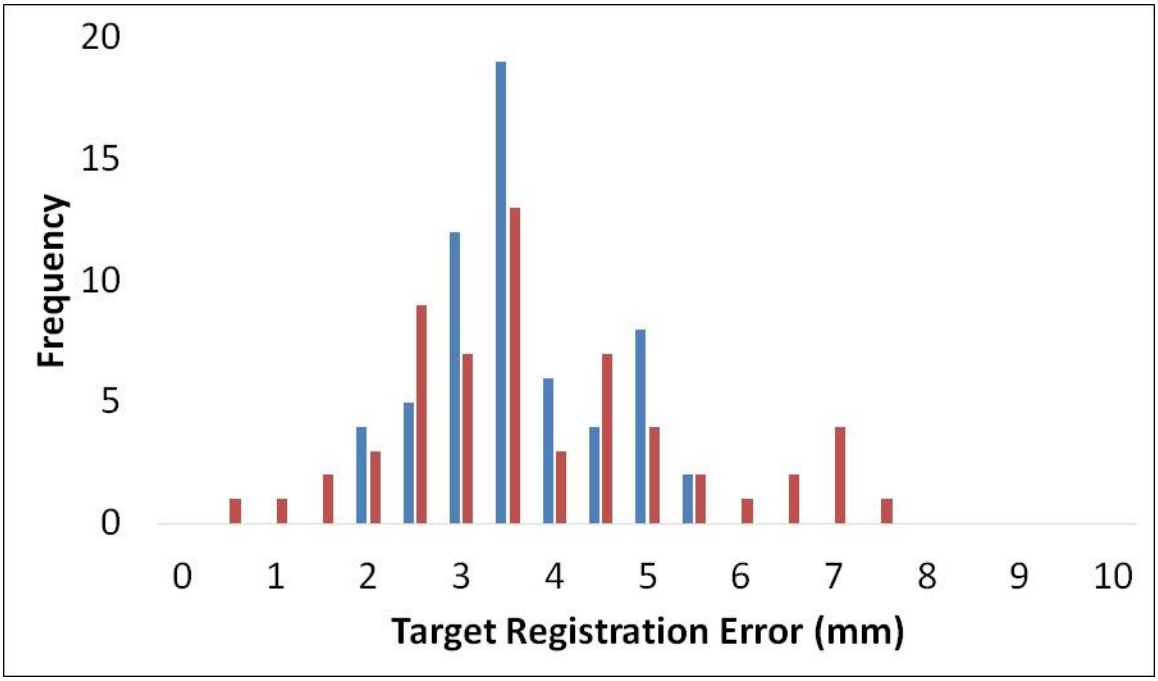

Figure 4.13: Histogram of target registration error values for each target point for each algorithm for registration of the reconstructed Targeting Tutor phantom volume. Point-set registration is indicated in red and linear object registration is indicated in blue.

Fiducial registration error for the point-set registration was 1.43mm on average compared to 1.15mm on average for linear object registration.

No precise temporal performance results are available for this experiment as the open-source 3D Slicer implementation was used to calculated the registrations. It was noted, though, that all linear object registrations took less than one second (38 collected points on average).

4.6 Discussion of Validation Results

One interesting observation is that for all validation experiments in which both point-set registration and linear object registration were performed, the average feature registration error was smaller for linear object registration than point-set registration. Though, it is noted that Fitzpatrick *et al.* [8] demonstrate that this is an unreliable metric for evaluating the quality of registration. It is further observed that the feature localization error is expected to be lower for linear object registration because lines and planes are constrained in fewer dimensions than

points. Fiducial registration error remains an important statistic, however, because it can also be shown that target registration error decreases with number of collected points, given the same feature registration error. For linear object registration where thousands of points were collected, this implies a highly accurate registration.

All registrations were performed using a Matlab implementation of the proposed algorithm, except for the Targeting Tutor phantom volume registrations, which were performed using the current open-source 3D Slicer implementation. Since registration is typically performed offline, the reported temporal performances are sufficient for practical use. This contrasts with the work from Olsson *et al.* [31], which has computational times up to 30s with fewer than 20 collected points. For datasets with greater than 10,000 collected points, such as the phantom registration data presented above, their exponential time algorithm may take impractically long to run.

Notably, for the phantom to model registration experiments, the rotational error was significantly larger for the fCal phantom and the translational error was significantly larger for the lumbar spine phantom. It is hypothesized that these errors are due strictly to the differing landmark point and linear object configurations for the two phantoms, though no rigorous study has been performed to test this hypothesis. Importantly, the error metrics are not significantly larger than the manufacturer reported tracking error, so the registration results are acceptable.

For the gel block targeting tool registration results, the average translational difference between the mean point-set registration and the linear object registrations was larger than the manufacturer reported tracking error. It was, however, smaller than the fiducial registration error. Fitzpatrick *et al.* [8] show that the fiducial registration error should always be smaller than the fiducial localization error (i.e. the tracking error), and thus, it is hypothesized that the tracking error in the validation experiments is in practice larger than the tracking error reported by the manufacturer. This may be due to different lighting conditions or preparation of markers for the

optical tracker. Thus, it can be concluded that the similarity metrics for this experiment are smaller than the practical tracking error and linear object registration is sufficiently accurate.

Though the registration procedure for the phantom to model and phantom to volume registrations follow a similar workflow, these represent two distinct applications of the linear object registration algorithm. In the phantom to model registrations, the computerized model is used to 3D print a physical phantom, whereas in the phantom to volume registration, a physical object is imaged and its image is segmented into a computerized model. In terms of validation, this distinction is important for two reasons. First, the sources and magnitudes of error were different from the two experiments. The accuracy associated with 3D printing was 0.50mm and affected manufacturing of the physical object, while the resolution of the CT imaging was 0.63mm and affected the image segmentation. Additionally, for the phantom to model registrations, linear objects were collected on the computerized model using the methods in subsection 2.4.3, whereas in the phantom to volume registrations, linear objects were collected on the segmented image using the methods in subsection 2.4.2 (due to noise in the imaging).

Chapter 5

CONCLUSION

5.1 Summary of Work

In this work, a linear object registration algorithm for simultaneous registration of points, lines, and planes was developed, implemented, and tested. This algorithm offers a registration solution for image-guided surgeries when the object to be registered does not have fiducial points. The algorithm satisfies the following criteria, which makes it feasible for use as a medical computing registration tool:

- Finds correspondence between linear objects
- Finds a unique solution for any sufficient configuration of linear objects
- Approaches the global optimum as the noise amplitude approaches zero
- Runs in polynomial time

The algorithm proceeds be the following workflow:

1. Determine correspondences between linear objects.
2. Compute the "linear object centroid", an invariant feature, for both set of linear objects.
3. Use the "linear object centroid" to determine points for point-set registration.
4. Compute point-set registration.

Additionally, the algorithm is optimized for medical computing applications by providing methods for extracting linear object from medical data sources (including tracked surgical tools and medical images) and promoting convergence for medical datasets using an iterative algorithm.

The proposed algorithm was validated for several applications, including various phantom registration applications and volume registrations. From the validation studies, several conclusions were drawn.

For phantom registration when fiducials were available for point-set registration, the linear object registration was insignificantly different from the point-set registration. For the tracked phantom to model registrations, the tracking accuracy was 1.4mm, while the translational difference in registrations were 0.74mm and 1.15mm for the fCal ultrasound calibration phantom and lumbar spine phantom respectively. For the tracked phantom to volume registrations, the translational difference in registrations was 0.74mm compared to 0.92mm fiducial registration error.

For phantom registration when fiducials were not available for point-set registration, the linear object registration produced better registration accuracy metrics. This indicates linear object registration performs better when fiducials are not available. For the LEGO® brick ultrasound calibration phantom, the target registration error and point reconstruction accuracies were both smaller for the linear object registration compared to point-set registration (1.18mm vs. 1.34mm and 3.46mm vs 3.98mm, respectively).

For reconstructed volume registration, linear object registration exhibited slightly better accuracy than the point-set registration, but significantly better precision (mean: 3.87mm, standard deviation: 0.91mm vs. mean: 4.05mm, standard deviation: 1.57mm, respectively). This demonstrates feasibility of linear object registration for volume registration.

Temporal performance of the algorithm was also found to be practically acceptable. The longest average registration time was 75s (for the lumbar spine phantom registration using tracked tool data). This registration used over 10,000 points. This temporal performance is adequate for practical scenarios where registration is performed offline.

Overall, the proposed linear object registration algorithm produces results that merit its use in several medical computing registration problems when fiducial points are not available. To this end, the algorithm was implemented as a 3D Slicer (www.slicer.org) module in the SlicerIGT (www.slicerigt.org) extension. This provides a usable tool for a multitude of registration applications when fiducial points are not available.

5.2 Future Work

To further the usability of linear object registration, several avenues of future work should be considered.

First, the way in which linear object configuration affects the registration result should be investigated. Fitzpatrick *et al.* [8] have studied this for point-set registration and determined several theoretical results. In particular, they note that target registration error is minimized at the centroid of the fiducial configuration. This result may imply that linear objects should be collected to encompass the relevant anatomy, but their result should be rigorously extended to linear object registration.

Although it has been shown that correct matching cannot be guaranteed without some construct of "references" (or another external feature to the linear object sets), the current "references" are cumbersome to collect and require proper placement by the user. To improve the algorithm's usability, it may be possible to automatically suggest reference locations to the user.

Because ultrasound volume reconstruction can be imprecise, further validation of the proposed algorithm should be performed on medical image volume registration. This would further demonstrate the algorithm's applicability to a wide variety of medical computing applications. Registration of volumetric images (for example, CT or MRI) of other surgical tools, robots, or phantoms (for example, Z-frame or N-wire phantoms) should be performed. These

applications offer a more practical scenario under which linear object registration for volumetric images may be used.

Finally, there exist a vast array of surgical tools which are not manufactured with well-defined linear objects. The proposed algorithm could be extended to handle parametric objects (or other classes of objects), for example, spheres, cylinders, or conic sections. This, however, may require significant modifications to the current algorithm, which uses several properties that are unique to linear objects.

Bibliography

[1] T. Ungi, D. Sargent, E. Moult, A. Lasso, C. Pinter, R. C. McGraw and G. Fichtinger, "Perk Tutor: An open-source training platform for ultrasound-guided needle insertions," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 12, pp. 3475-3481, 2012.

[2] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3D Point-Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 698-700, 1987.

[3] B. K. P. Horn, "Closed form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, pp. 629-642, 1987.

[4] A. Muns, C. Muhl, R. Hasse, H. Mockel, C. Chalopin, J. Meixensberger and D. Linder, "A neurosurgical phantom-based training System with Ultrasound simulation," *European Journal of Neurosurgery*, 2013.

[5] R. Shamir, M. Freiman, L. J. amd M Shoham, E. Zehavi and Y. Shoshan, "Robot-Assisted Image-Guided Targeting for Minimally Invasive Neurosurgery: Planning, Registration, and In-vitro Experiment," in *Medical Image Computing and Computer Assisted Intervention*, 2005.

[6] R. Walsh, M. Soehl, A. Rankin, A. Lasso and G. Fichtinger, "Design of a tracked ultrasound calibration made of LEGO bricks," in *SPIE Medical Imaging*, 2014.

[7] P. H. Schonemann, "A Generalized Solution of the Orthogonal Procrustes Problem," *Psychometrika*, vol. 31, pp. 1-10, 1966.

[8] J. M. Fitzpatrick, J. B. West and C. R. Maurer, "Predicting Error in Rigid-Body Registration," *IEEE Transactions on Medical Imaging*, vol. 17, pp. 694-702, 1998.

[9] J. M. Fitzpatrick and J. B. West, "The Distribution of Target Registration Error in Rigid-Body Point-Based Registration," *IEEE Transactions on Medical Imaging*, vol. 20, no. 9, pp. 917-927, 2001.

[10] A. D. Wiles, A. Likholyot, D. D. Frantz and T. M. Peters, "A Statistical Model for Point-Based Target Registration Error With Anisotropic Fiducial Localizer Error," *IEEE Transactions on Medical Imaging*, vol. 27, no. 3, pp. 378-390, 2008.

[11] M. Moghari and P. Abolmaesumi, "Distribution of Target Registration Error for Anisotropic and Inhomogeneous Fiducial Localization Error," *IEEE Transactions on Medical Imaging,* vol. 28, no. 6, pp. 799-813, 2009.

[12] A. Danilchenko and J. M. Fitzpatrick, "General Approach to First-Order Error Prediction in Rigid Point Registration," *IEEE Transactions on Medical Imaging,* vol. 30, no. 3, pp. 679-693, 2011.

[13] M. Moghari and P. Abolmaesumi, "Understanding the Effect of Bias in Fiducial Localization Error on Point-Based Rigid-Body Registration," *IEEE Transactions on Medical Imaging,* vol. 29, no. 10, pp. 1730-1738, 2010.

[14] J. M. Fitzpatrick, "Fiducial Registration Error and Target Registration Error are Uncorrelated," in *SPIE Medical Imaging*, 2009.

[15] P. J. Besl and N. D. McKay, "A Method for Registration of 3D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 14, pp. 239-256, 1992.

[16] Y. Chen and G. Medioni, "Object Modeling by Registration of Multiple Range Images," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1991.

[17] S. Rusinkiewicz and M. Levoy, "Efficient Variants of the ICP Algorithm," in *IEEE 3D Digital Imaging and Modeling*, 2001.

[18] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu and E. Mjolsness, "New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence," *Pattern Recognition,* vol. 31, no. 8, pp. 1019-1031, 1998.

[19] M. Sofka, G. Yang and C. V. Stewart, "Simultaneous Covariance Driven Correspondence (CDC) and Transformation Estimation in the Expectation Maximization Framework," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[20] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 12, pp. 2262-2275, 2010.

[21] P. Foroughi, R. Taylor and G. Fichtinger, "Revisited Initialization for 3D Bone Registration," in *Proceedings of SPIE*, 2008.

[22] H. Li and R. Hartley, "The 3D-3D Registration Problem Revisited," in *IEEE International Conference on Computer Vision*, 2007.

[23] A. W. Fitzgibbon, "Robust Registration of 2D and 3D Point Sets," *Image and Vision Computing,* vol. 21, pp. 1145-1153, 2003.

[24] J. P. Thirion, "Extremal Points: Definition and Application to 3D Image Registration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994.

[25] G. Xiao, S. H. Ong and K. W. C. Foong, "Efficient Partial-Surface Registration for 3D Objects," *Computer Vision and Image Understanding,* vol. 98, pp. 271-293, 2005.

[26] 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.

[27] B. Jian and B. C. Vemuri, "A Robust Algorithm for Point Set Registration Using Mixture of Gaussians," in *IEEE Conference on Computer Vision*, 2005.

[28] S. Lee, G. Fichtinger and G. S. Chirikjian, "Numerical algorithms for spatial registration of line fiducial from cross-sectional images," *Medical Physics,* vol. 29, no. 8, pp. 1881-1891, 2002.

[29] A. K. Jain, T. Mustafa, Y. Zhou, C. Burdette, G. S. Chirijian and G. Fichtinger, "FTRAC - A robust fluoroscopic tracking fiducial," *Medical Physics,* vol. 32, no. 10, pp. 3185-3198, 2005.

[30] C. R. Meyer, G. S. Leichtman, J. A. Brunberg, R. L. Wahl and L. E. Quint, "Simultaneous Usage of Homologous Points, Lines, and Planes for Optimal 3D, Linear Registration of Multimodality Imaging Data," *IEEE Transactions on Medical Imaging,* vol. 14, pp. 1-11, 1995.

[31] C. Olsson, F. Kahl and M. Oskarsson, "Branch-and-Bound Methods for Euclidean Registration Problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, no. 5, pp. 783-794, 2009.

[32] A. Lasso, T. Heffter, A. Rankin, C. Pinter, T. Ungi and G. Fichtinger, "PLUS: Open-Source Toolkit for Ultrasound-Guided Intervention Systems," *IEEE Transactions on Biomedical Engineering,* 05/2014 2014.

[33] T. K. Chen, A. D. Thurston, R. E. Ellis and P. Abolmaesumi, "A Real-Time Freehand Ultrasound Calibration System with Automatic Accuracy Feedback and Control," *Ultrasound in Medicine \& Biology,* vol. 35, no. 1, pp. 79-93, 2009.

[34] M. Anand, F. King, T. Ungi, A. Lasso, J. Rudan, J. Jagadeesan, J. Fritz, F. Jolesz, J. Carrino and G. Fichtinger, "Design and Development of a Mobile Image Overlay System for Image-Guided Needle Interventions," in *36th Annual International Conference of IEEE EMBS*, 2014.

[35] T. Ungi, D. Beiko, M. Fuoco, F. King, M. S. Holden, G. Fichtinger and D. R. Siemens, "Tracked Ultrasonography Snapshots Enhance Needle Guidance for Percutaneous Renal Access: A Pilot Study," *Journal of Endourology*, 50/2014 2014.

[36] G. Carbajal, A. Lasso, A. Gomez and G. Fichtinger, "Improving N-wire phantom-based freehand ultrasound calibration," *International Journal of Computer Assisted Radiology and Surgery*, vol. 8, pp. 1063-1072, 2013.

[37] M. Welch, J. Andrea, T. Ungi and G. Fichtinger, "Freehand Ultrasound Calibration: Phantom Versus Tracked Pointer," in *SPIE Medical Imaging*, 2013.

[38] T. Vaughan, A. Lasso and G. Fichtinger, "Ultrasound Volume Reconstruction: Open-Source Implementation with Hole Filling Functionality," in *5th NCIGT and NIH Image Guided Therapy Workshop*, 2012.

Appendix A
ALGORITHMIC PROOFS

Proof of Construction of the Estimator $\hat{\varphi}$

The objective is to show that $\forall a \epsilon S_A, \hat{\varphi}(a) = \varphi(a)$. To this end, suppose that $\hat{\varphi}(a) \neq \varphi(a)$.

Case 1: Suppose that $\hat{\varphi}(a) = b \epsilon S_B$.

Since $\hat{\varphi}(a) = b \epsilon S_B$, this implies that $dist\big(\rho(a), \rho(b)\big) < \delta$.

$$\Rightarrow \exists a \epsilon S_A, \exists b \epsilon S_B | dist\big(\rho(a), \rho(b)\big) < \delta \wedge \varphi(a) \neq b$$

This contradicts the requirements for ρ and δ. Thus, this implies that for any $\hat{\varphi}(a) = b \epsilon S_B, \hat{\varphi}(a) = \varphi(a)$.

Case 2: Suppose that $\hat{\varphi}(a) \notin S_B$ and $\varphi(a) = b \in S_B$.

Since $\varphi(a) = b \in S_B$, this implies

$$\Rightarrow dist\big(\rho(a), \rho(b)\big) < \delta \wedge \forall \hat{b} \neq b \in S_B, dist\left(\rho(a), \rho\big(\hat{b}\big)\right) > \delta$$

But this implies that $\min\limits_{b \epsilon S_B}\big[dist\big(\rho(a), \rho(b)\big)\big] < \delta$, which by definition of $\hat{\varphi}(a)$, requires $\hat{\varphi}(a) = \operatorname{argmin}\limits_{b \epsilon S_B}\big[dist\big(\rho(a), \rho(b)\big)\big] = b \in S_B$. But this contradicts $\hat{\varphi}(a) \notin S_B$. Thus, if $\hat{\varphi}(a) \notin S_B$ then $\varphi(a) \notin S_B$.

Thus, we get $\forall a \epsilon S_A, \hat{\varphi}(a) = \varphi(a)$.

Proof of Construction of the Relaxation of the Estimator $\hat{\varphi}$

The objective is to show that $\forall a \epsilon S_A, \hat{\varphi}(a) = \varphi(a)$. To this end, suppose that $\hat{\varphi}(a) \neq \varphi(a)$.

Case 1: Suppose that $\hat{\varphi}(a) = b \epsilon S_B$.

Since $\hat{\varphi}(a) = b \epsilon S_B$, this implies that $dist\big(\rho(a), \rho(b)\big) < \delta$, since the matching would have been rejected otherwise.

$$\Rightarrow \exists a \epsilon S_A, \exists b \epsilon S_B | dist\big(\rho(a), \rho(b)\big) < \delta \wedge \varphi(a) \neq b$$

This contradicts the requirements for ρ and δ. Thus, this implies that for any $\hat{\varphi}(a) = b \epsilon S_B, \hat{\varphi}(a) = \varphi(a)$.

Case 2: Suppose that $\hat{\varphi}(a) \notin S_B$ and $\varphi(a) = b \in S_B$.

Since, $\varphi(a) = b \in S_B$, this implies

$$\Rightarrow \forall \hat{b} \neq b \in S_B, dist\left(\rho(a), \rho\left(\hat{b}\right)\right) > \delta \wedge \forall \hat{a} \neq a \in S_A, dist\left(\rho(\hat{a}), \rho(b)\right) > \delta$$

Thus, this implies that $\hat{\varphi}(a) = b$, is an optimal assignment, which would have been found be the Kuhn-Munkres algorithm. It would not have been rejected, since by $\varphi(a) = b \in S_B$, it is guaranteed that $dist\left(\rho(a), \rho(b)\right) < \delta$, thus, $\hat{\varphi}(a) = b$, which is a contradiction. Thus, if $\hat{\varphi}(a) \notin S_B$ then $\varphi(a) \notin S_B$.

Thus, we get $\forall a \epsilon S_A, \hat{\varphi}(a) = \varphi(a)$.

Proof that References can Completely Determine Linear Object Correspondence

Suppose that we choose M references and each reference is collected with associated noise ε.

Now, suppose for the entire set of linear objects in both the A and B coordinate frames, the following holds:

$$\forall a \in S_A, \exists \vec{x} \in a | \forall \hat{a} \neq a \in S_A, dist(\vec{x}, \hat{a}) > 4\varepsilon$$

$$\forall b \in S_B, \exists \vec{x} \in b | \forall \hat{b} \neq b \in S_B, dist\left(\vec{x}, \hat{b}\right) > 4\varepsilon$$

This condition imposes the condition that for any pair of linear objects in the same coordinate frame, there exists at least one point on each linear object that is some threshold distance from the other linear object. This condition is already naturally implied by collecting distinct linear objects and is, thus, not unreasonable to impose in practice.

This implies, given a known transformation $T_{A \to B}$ (and conversely $T_{B \to A}$)

$$\forall a \in S_A, \exists T_{A \to B}\vec{x} \in \varphi(a) | \forall \hat{b} \neq \varphi(a) \in S_B, dist\left(T_{A \to B}\vec{x}, \hat{b}\right) > 4\varepsilon$$

$$\forall b \in S_B, \exists T_{B \to A}\vec{x} \in \varphi^{-1}(b) | \forall \hat{a} \neq \varphi^{-1}(b) \in S_A, dist(T_{B \to A}\vec{x}, \hat{a}) > 4\varepsilon$$

Now, given a set of M references in coordinate frame X with positions $\vec{x}_{R,i} = \vec{x}'_{R,i} + \vec{\varepsilon}_{XR,i}$ (where $\left|\vec{\varepsilon}_{XR,i}\right| < \varepsilon$), the signature of an arbitrary linear object $x \in S_X$ can be decomposed as:

$$sig(x) = \begin{bmatrix} dist\left(x, \vec{x}_{R,1}\right) \\ \vdots \\ dist\left(x, \vec{x}_{R,M}\right) \end{bmatrix}$$

$$sig(x) = \begin{bmatrix} dist\left(x, \vec{x}'_{R,1} + \vec{\varepsilon}_{XR,1}\right) \\ \vdots \\ dist\left(x, \vec{x}'_{R,M} + \vec{\varepsilon}_{XR,M}\right) \end{bmatrix}$$

$$sig(x) = \begin{bmatrix} dist\left(x, \vec{x}'_{R,1}\right) \\ \vdots \\ dist\left(x, \vec{x}'_{R,M}\right) \end{bmatrix} + \begin{bmatrix} \varepsilon'_{XR,1} \\ \vdots \\ \varepsilon'_{XR,M} \end{bmatrix}$$

where it is guaranteed that $\forall i \in 1 \dots M, \left|\varepsilon'_{XR,i}\right| < \varepsilon$.

Now, for every linear object in the coordinate frame A, it can be guaranteed that:

$$dist\left(sig(a), sig\left(\varphi(a)\right)\right) = \left\| \begin{bmatrix} dist\left(a, \vec{a}'_{R,1}\right) \\ \vdots \\ dist\left(a, \vec{a}'_{R,M}\right) \end{bmatrix} + \begin{bmatrix} \varepsilon'_{AR,1} \\ \vdots \\ \varepsilon'_{AR,M} \end{bmatrix} - \begin{bmatrix} dist\left(\varphi(a), \vec{b}'_{R,1}\right) \\ \vdots \\ dist\left(\varphi(a), \vec{b}'_{R,M}\right) \end{bmatrix} - \begin{bmatrix} \varepsilon'_{BR,1} \\ \vdots \\ \varepsilon'_{BR,M} \end{bmatrix} \right\|$$

Since distances are preserved under rigid transformation, the following must hold:

$$dist\left(sig(a), sig\left(\varphi(a)\right)\right) = \left\| \begin{bmatrix} \varepsilon'_{AR,1} \\ \vdots \\ \varepsilon'_{AR,M} \end{bmatrix} - \begin{bmatrix} \varepsilon'_{BR,1} \\ \vdots \\ \varepsilon'_{BR,M} \end{bmatrix} \right\|$$

$$dist\left(sig(a), sig\left(\varphi(a)\right)\right) < \sqrt{\sum_{i=1}^{M} \left(\left|\varepsilon'_{AR,i}\right| + \left|\varepsilon'_{BR,i}\right|\right)^2}$$

$$dist\left(sig(a), sig\left(\varphi(a)\right)\right) < 2\sqrt{M}\varepsilon$$

Now, suppose that there exists an $a \in S_A$ and a $b \in S_B$ for which $\varphi(a) \neq b$, such that

$$dist\left(sig(a), sig(b)\right) \equiv D_{ab} < 2\sqrt{M}\varepsilon$$

Then, it is possible to add a total of K references each at some point $\vec{x} \in a | \forall \hat{a} \neq a \in S_A, dist(\vec{x}, \hat{a}) > 4\varepsilon + \eta$ for an arbitrarily small $\eta > 0$.

Given the new references, this yields

$$dist\left(sig(a), sig(b)\right)^2 = {D_{ab}}^2 + \sum_{k=1}^{K} \left(dist(\vec{x}_k, a) + \varepsilon'_{AR,M+k} - dist(T_{A\to B}\vec{x}_k, b) - \varepsilon'_{BR,M+k}\right)^2$$

$$dist\left(sig(a), sig(b)\right)^2 > {D_{ab}}^2 + \sum_{k=1}^{K} \left(\varepsilon'_{AR,M+k} - \varepsilon'_{BR,M+k} - 4\varepsilon - \eta\right)^2$$

$$dist\left(sig(a), sig(b)\right)^2 > {D_{ab}}^2 + \sum_{k=1}^{K} (2\varepsilon + \eta)^2$$

$$dist\left(sig(a), sig(b)\right)^2 > {D_{ab}}^2 + K(2\varepsilon + \eta)^2$$

So, K can be made arbitrarily large such that $dist\left(sig(a), sig(b)\right) > 2\sqrt{M + K}\varepsilon$. From the above, this condition is guaranteed to be true whenever:

$$D_{ab}{}^2 + K(2\varepsilon + \eta)^2 > \left(2\sqrt{M + K}\varepsilon\right)^2$$

$$K > \frac{4M\varepsilon^2 - D_{ab}{}^2}{2\varepsilon\eta + \eta^2}$$

This process may be repeated an arbitrary number of times until $\forall a \in S_A, dist\big(sig(a), sig(b)\big) < 2\sqrt{M + K}\varepsilon \Leftrightarrow \varphi(a) = b$.

Thus, it has been shown that it is always possibly to construct a set of references which completely and correctly specifies the linear object correspondence.

Proof Correspondence Cannot be Determined Completely by Linear Object Sets

Here, a proof by example is given to show that there exists a set of linear objects which admits a unique registration, but correspondence cannot be determined.

Consider the following set of points in coordinate frames A and B:

$$a_1 = \begin{bmatrix}0\\0\\0\end{bmatrix}, a_2 = \begin{bmatrix}1\\0\\0\end{bmatrix}, a_3 = \begin{bmatrix}1/2\\\sqrt{3}/2\\0\end{bmatrix}$$

$$b_1 = \begin{bmatrix}0\\0\\0\end{bmatrix}, b_2 = \begin{bmatrix}1\\0\\0\end{bmatrix}, b_3 = \begin{bmatrix}1/2\\\sqrt{3}/2\\0\end{bmatrix}$$

One solution to the registration problem is the trivial solution $T_{A\to B} = I$. In fact, because the points are not collinear, this is the only solution. The fiducial registration error is zero.

Now, consider relabeling the points in coordinate frame B:

$$b_2 = \begin{bmatrix}0\\0\\0\end{bmatrix}, b_3 = \begin{bmatrix}1\\0\\0\end{bmatrix}, b_1 = \begin{bmatrix}1/2\\\sqrt{3}/2\\0\end{bmatrix}$$

Now, using a known point-set registration algorithm, it can be calculated that the registration result is:

$$T_{A\to B} = \begin{bmatrix}-1/2 & \sqrt{3}/2 & 0 & 1/2\\-\sqrt{3}/2 & -1/2 & 0 & \sqrt{3}/2\\0 & 0 & 1 & 0\\0 & 0 & 0 & 1\end{bmatrix}$$

Again, because the points are not collinear, this is the only solution. Furthermore, the fiducial registration error is still zero.

This yields the conclusion that based only on the two available point-sets, there is no unique labelling that yields a better registration result than all others, so it is not possible to determine labelling without further information.

This result does not preclude the existence of a set of linear objects for which the correspondence can be uniquely determined from the set of linear object itself. Rather, this supports the notion of references, which can be used to determine correspondence, being necessary to determine linear object correspondence for an arbitrary set of linear objects.

Proof of Existence and Uniqueness of Linear Object Centroid

Points, lines, and planes can be defined by a set of normal vectors ($\vec{n}_1, \vec{n}_2, \vec{n}_3$) and a single point ($\vec{b}$). This representation easily lends itself to calculating the distance to a point.

The squared distance from a point $\vec{x}$ to a point is:

$$\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_1\right]^2+\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_2\right]^2+\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_3\right]^2$$

The squared distance from a point $\vec{x}$ to a line is:

$$\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_1\right]^2+\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_2\right]^2$$

The squared distance from a point $\vec{x}$ to a plane is:

$$\left[\left(\vec{x}-\vec{b}\right)^T\vec{n}_1\right]^2$$

The centroid of a set of points, lines, and planes can be defined as the point $\vec{x}$, which minimizes the sum of squared distances to all linear objects. This can be formulated as minimizing the objective function f:

$$f=\sum_{Point}\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,1}\right]^2+\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,2}\right]^2+\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,3}\right]^2$$
$$+\sum_{Line}\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,1}\right]^2+\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,2}\right]^2+\sum_{Plane}\left[\left(\vec{x}-\vec{b}_i\right)^T\vec{n}_{i,1}\right]^2$$

It is well-known that this can be reformulated as the least-squares solution (assuming that $\vec{n}$ and $\vec{b}$ are indexed appropriately over a total of m linear objects):

$$N\vec{x}=B$$

$$N = \begin{bmatrix} \vec{n}_1{}^T \\ \vdots \\ \vec{n}_m{}^T \end{bmatrix}$$

$$B = \begin{bmatrix} \vec{b}_1{}^T \vec{n}_1 \\ \vdots \\ \vec{b}_m{}^T \vec{n}_m \end{bmatrix}$$

The least squares solution is well known, and guaranteed to exist:

$$\vec{x} = (N^T N)^{-1} N^T B$$

Suppose the solution to this equation is not unique. In particular, suppose that there exist two solutions $\vec{x}_1 \neq \vec{x}_2$, satisfying the solution equation. It can be shown that this implies

$$N(\vec{x}_1 - \vec{x}_2) = \vec{0}$$

Defining $\vec{d} = (\vec{x}_1 - \vec{x}_2) \neq \vec{0}$, we get $N\vec{d} = \vec{0}$ (i.e. $\forall i, \vec{n}_i \cdot \vec{d} = 0$).

Now, consider two sets S_A and S_B of unregistered linear objects that have known correspondence. Suppose that the optimal registration is given by some transformation T such that for each point $\vec{x}$ on a linear object in set S_A the corresponding point is $\vec{y}$ on set S_B.

$$D(\vec{y}, T\vec{x}) = (\vec{y} - T\vec{x})^T (\vec{y} - T\vec{x})$$

Now, consider the registration T' where $T'\vec{x} = T\vec{x} + \vec{d}$ where $\vec{d} \neq \vec{0}$ and $\vec{n}_i \cdot \vec{d} = 0$ for all i over all linear objects defined in set S_B. Any linear object in the set can be defined by the equation $\forall j, (\vec{y} - \vec{b}) \cdot \vec{n}_j = 0$ (where j indexes over all normal vector associated with the linear object). Since $\forall i, \vec{n}_i \cdot \vec{d} = 0$, it is also true that $\forall j, (\vec{y} + \vec{d} - \vec{b}) \cdot \vec{n}_j = 0$. Thus, for any point $\vec{y}$ that lies on the linear object, $\vec{y} + \vec{d}$ must also lie on the linear object.

$$D(\vec{y} + \vec{d}, T'\vec{x}) = D(\vec{y}, T\vec{x})$$

Therefore, if the centroid is not unique then the sum of squares distances for both registrations is equal, which implies that the registration is not unique.

Proof of Invariance of Linear Object Centroid

Given a set of linear objects S_A and a set of linear objects S_B with known correspondence, points $\vec{x}$ in the set S_A can be transformed to corresponding points in set S_B by the transformation $T\vec{x} = R\vec{x} + \vec{d}$, where R is a pure rotation for which we observe that $R^T = R^{-1}$ and $\vec{d}$ is a pure translation. Now, consider any normal vector in the set S_A. The normal vector can be defined by

two points in the frame of A, say $\vec{n} = \vec{x}_1 - \vec{x}_2$. The transformed normal vector is rotated but not translated.

$$T\vec{n} = R\vec{n}$$

We note specifically the result $\vec{n}_B = R\vec{n}_A$. Now, consider the centroid of the set S_A.

$$\vec{c}_A = (N_A^T N_A)^{-1} N_A^T B_A$$

Transforming the centroid by the known transformation T yields

$$T\vec{c}_A = R(N_A^T N_A)^{-1} N_A^T B_A + \vec{d}$$

$$T\vec{c}_A = \left(N_B{}^T N_B\right)^{-1} N_B{}^T B_A + \vec{d}$$

Now, consider the matrix B_A separately, where it is noted that the linear objects in the B coordinate frame can be written with $\vec{b}_B = T\vec{b}_A = R\vec{b}_A + \vec{d}$.

$$B_A = \begin{bmatrix} \vec{b}_{1,A}{}^T \vec{n}_{1,A} \\ \vdots \\ \vec{b}_{m,A}{}^T \vec{n}_{m,A} \end{bmatrix}$$

$$B_A = \begin{bmatrix} \vec{b}_{1,B}{}^T \vec{n}_{1,B} \\ \vdots \\ \vec{b}_{m,B}{}^T \vec{n}_{m,B} \end{bmatrix} - \begin{bmatrix} \vec{d}^T \vec{n}_{1,B} \\ \vdots \\ \vec{d}^T \vec{n}_{m,B} \end{bmatrix}$$

$$B_A = B_B - N_B \vec{d}$$

Now, the two above results can be combined and simplified to show:

$$T\vec{c}_A = \vec{c}_B$$

Thus, the centroid for the set of linear objects B is the transformed centroid from set S_A, demonstrating the invariance of the centroid under transformation.

Additionally, the projections of the centroid onto each linear object are invariant under transformation. In any coordinate system, the projection of a single point $\vec{c}$ onto a linear object can be defined.

$$\vec{p} = \vec{c} - \sum_{j \in Normals} \vec{n}_{i,j} \left(\vec{c} - \vec{b}_i\right)^T \vec{n}_{i,j}$$

The projection of the centroid in set A onto an arbitrary linear object in set S_A.

$$T\vec{p}_A = R\vec{c}_A + \vec{d} - \sum_{Normals} R\vec{n}_{i,j,A} \left(\vec{c}_A - \vec{b}_{i,A}\right)^T \vec{n}_{i,j,A}$$

$$T\vec{p}_A = R\vec{c}_A + \vec{d} - \sum_{Normals} R\vec{n}_{i,j,A} \left(R\vec{c}_A + \vec{d} - R\vec{b}_{i,A} - \vec{d}\right)^T R\vec{n}_{i,j,A}$$

$$T\vec{p}_A = \vec{p}_B$$

Thus, the projection of the centroid onto each of the linear objects is invariant under the transformation. In fact, no specials properties of the centroid were used, and thus, the projection of any invariant point is invariant under transformation.

Additionally, if the sets A and B are all points on all linear objects in the A and B coordinate systems respectively.

$$B = TA$$

For any point $\vec{c}_A$ and its corresponding point $\vec{c}_B = T\vec{c}_A = R\vec{c}_A + \vec{d}$.

$$B - \vec{c}_B = R(A - \vec{c}_A)$$

Thus, the centroid (or again, any other invariant) can be used to reduce the 6DOF registration problem to a 3DOF spherical registration problem.

Proof that Points Chosen for Point-Set Registration are Sufficient

The set of centroid projections and direction/normal vectors is used for spherical registration. Suppose that these points are all collinear along some line intersecting the origin. The registration would fail only in this case. Then each point on the linear object can be written as $k_i\vec{u}$ (with i appropriately indexing the linear objects) and each direction/normal vector can be written as $\vec{u}$.

Points: $\vec{x} = k_i\vec{u}$

Lines: $\vec{x} = k_i\vec{u} + t\vec{u}$

Planes: $(\vec{x} - k_i\vec{u})^T\vec{u} = 0$

Now, consider two sets S_A and S_B of unregistered linear objects that have known correspondence. Suppose that the optimal registration is given by some transformation T such that for each point $\vec{x}$ on a linear object in set S_A the corresponding point is $\vec{y}$ in set S_B.

$$D(\vec{y}, T\vec{x}) = (\vec{y} - T\vec{x})^T(\vec{y} - T\vec{x})$$

Now, consider the registration T' where $T'\vec{x} = R_u T\vec{x}$ where R_u is a pure rotation about the axis $\vec{u}$.

Consider $\vec{y}$ on a point.

$$R_u\vec{y} = R_u k_i\vec{u}$$

$$R_u\vec{y} = \vec{y}$$

Consider $\vec{y}$ on a line.

$$R_u\vec{y} = R_u k_i\vec{u} + R_u t\vec{u}$$

$$R_u\vec{y} = \vec{y}$$

Consider $\vec{y}$ on a plane.

$$(\vec{y} - k_i\vec{u})^T\vec{u} = 0$$

$$(R_u\vec{y} - k_i\vec{u})^T\vec{u} = 0$$

Thus, we have shown, that given a point on a point, line, or plane, rotation about the $\vec{u}$ axis maps this point to another point on the same point, line, or plane.

Now, suppose that under transformation T' for each point $\vec{x}$ on a linear object in set S_A the corresponding point is $\vec{y}'$ in set S_B. Thus, for any point $\vec{y}$ that lies on a linear object, $R_u\vec{y}$ must also lie on that linear object.

$$D(R_u\vec{y}, T'\vec{x}) = D(\vec{y}, T\vec{x})$$

Therefore, if the all points are collinear along a lines intersecting the origin, the sum of squares distances for both registrations is equal, which implies that the registration is not unique.

Suppose that the line of collinearity does not intersect the origin. Let $\vec{u}$ be the direction vector of the line of collinearity and $\vec{a}$ be the closest point on the line of collinearity to the origin ($\vec{a} \cdot \vec{u} = 0$). For each projection of the centroid onto a linear object $\vec{p}_i = \vec{a} + k_i\vec{u}$, it is possible to calculate its squared distance to the origin (we observe that every non-empty set of projections and direction/normal vectors must have at least one projection).

$$D^2(O, \vec{x}_i) = |\vec{a}|^2 + {k_i}^2|\vec{u}|^2$$

Additionally, the distance from the projection of the centroid to the closest point to the origin can be calculated:

$$D^2(\vec{a}, \vec{p}_i) = {k_i}^2|\vec{u}|^2$$

The sum of squares distances from the projection of the centroid to both the origin and the point $\vec{a}$ can be calculated.

$$\sum_{Objects} D^2(O, \vec{p}_i) = \sum_{Objects} |\vec{a}_i|^2 + \sum_{Objects} k^2|\vec{u}|^2 > \sum_{Objects} k^2|\vec{u}|^2 = \sum_{Objects} D^2(\vec{a}, \vec{x}_i)$$

This shows that the sum of squared distances to the point $\vec{a}$ is less than the sum of squared distances to the origin. Thus, the centroid of the set of points is not at the origin. This contradicts the supposition that the centroid is at the origin and the problem had been reduced to a pure spherical registration problem. Thus, the line of collinearity must intersect the origin.

Proof that Points Chosen for Point-Set Registration Yield Accurate Registration

Consider the Arun *et al.* [2] method of spherical registration for points with known correspondence. The data matrix in this method is found by taking for each point $\vec{x}'$, its

corresponding point $\vec{y}'$ and calculating the outer product of these two vectors. The data matrix is the sum over all points of $\vec{x}'_i \vec{y}'^T_i$.

$$H = \sum_{Points} \vec{x}'_i \vec{y}'^T_i = [\vec{x}'_1 \quad \cdots \quad \vec{x}'_m] \begin{bmatrix} \vec{y}'^T_1 \\ \vdots \\ \vec{y}'^T_m \end{bmatrix}$$

Now, if there are corresponding linear objects which are not necessarily points, to register all points on a linear object would be to sum over all parameter values given corresponding parameterizations of the objects. In the limit of an infinite number of points on each linear object, the sum becomes integration over parameters t.

$$H = \sum_{Objects} \int_{C_1}^{D_1} \cdots \int_{C_k}^{D_k} \overrightarrow{LO_A(t_1, \ldots, t_k)} \overrightarrow{LO_B(t_1, \ldots, t_k)}^T \, dt_k \cdots dt_1$$

Of course, registration of points is a trivial case in which the term can be added traditionally to the data matrix. Additionally, it is noted that a plane with unknown rotation about its normal can be equivalently expressed as its line normal to the plane, which passes through the same known point. Thus, points are trivially registered and planes are registered using their normal lines, so all that must be determined is the correct registration of lines.

$$\int_{-C}^{C} \left(\vec{b}_A + t\vec{v}_A\right)\left(\vec{b}_B + t\vec{v}_B\right)^T dt$$

$$= 2C\vec{b}_A\vec{b}_B{}^T + \frac{2C^3}{3}\vec{v}_A\vec{v}_B{}^T$$

Thus, registering lines by their closest point to the centroid and normal is equivalent to registering lines of length $\sqrt{3}$ in each direction from the projection of the centroid with weighting of $\frac{1}{2\sqrt{3}}$ relative to points.

Thus, the proposed algorithm solves registration of finite points, lines, and planes. Thus, it produces an accurate approximation to the registration of infinite linear objects.