NEW ALGORITHMS IN RIGID-BODY REGISTRATION AND ESTIMATION OF REGISTRATION ACCURACY

by

HEDJAZI MOGHARI MEHDI

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Doctor of Philosophy

Queen’s University
Kingston, Ontario, Canada
September 2008

Copyright © Hedjazi Moghari Mehdi, 2008
Abstract

Rigid-body registration is an important research area with major applications in computer-assisted and image-guided surgery. In these surgeries, often the relationship between the preoperative and intraoperative images taken from a patient must be established. This relationship is computed through a registration process, which finds a set of transformation parameters that maps some point fiducials measured on a patient anatomy to a preoperative model. Due to point measurement error caused by medical measurement instruments, the estimated registration parameters are imperfect and this reduces the accuracy of the performed registrations. Medical measurement instruments often perturb the collected points from the patient anatomy by heterogeneous noise. If the noise characteristics are known, they can be incorporated in the registration algorithm in order to more reliably and accurately estimate the registration parameters and their variances.

Current techniques employed in rigid-body registration are primarily based on the well-known Iterative Closest Points (ICP) algorithm. Such techniques are susceptible to the existence of noise in the data sets, and are also very sensitive to the initial alignment errors. Also, the literature offers no analytical solution on how to estimate the accuracy of the performed registrations in the presence of heterogeneous noise.
In an effort to alleviate these problems, we propose and validate various novel registration techniques based on the Unscented Kalman Filter (UKF) algorithm. This filter is generally employed for analyzing nonlinear systems corrupted by additive heterogenous Gaussian noise. First, we propose a new registration algorithm to fit two data sets in the presence of arbitrary Gaussian noise, when the corresponding points between the two data sets are assumed to be known. Next, we extend this algorithm to perform surface-based registration, where point correspondences are not available, but the data sets are roughly aligned. A solution to multi-body point and surface-based registration problem is then proposed based on the UKF algorithm.

The outputs of the proposed UKF registration algorithms are then utilized to estimate the accuracy of the performed registration. For the first time, novel derivations are presented that can estimate the distribution of registration error at a target in the presence of an arbitrary Gaussian noise.
Acknowledgments

First, I would like to deeply thank my parents (Hossein & Fati) and my sister (Mona) who have been my major supporters and encouragers while I have been abroad. Also, I would like to especially thank Professors Mehrdad Abedi and Seyed Mohammad Ahadi, my B.Sc. advisors, and Professor Abolghasem Raie, my M.Sc. advisor, in the Department of Electrical and Computer Engineering at Amirkabir University of Technology for encouraging and helping me to study my Ph.D. at Queen’s University in Canada. I would like to thank Professors Majid Ahmadi and Behnam Shahrrava in the Department of Electrical and Computer Engineering at the University of Windsor for accepting me as a research assistant and helping me come to Canada.

My especial thanks to Professor Purang Abolmaesumi, my Ph.D. supervisor, who accepted me as one of his students. Without his valuable support and guidance at every stage, this work would not have been possible. I also wish to thank Professor Armand from Johns Hopkins University, and Dr. Kunz and Dr. Beek from Kingston General Hospital for providing me with medical data sets. Finally, I wish to thank Dr. Ma, Mr. Chen and Mr. Tahmasebi for their collaboration and joint publications, and Miss Bailey who helped me with my writing style.

Mehdi Hedjazi Moghari

September 26, 2008.
# Table of Contents

Abstract i  
Acknowledgments iii  
Table of Contents iv  
List of Tables vii  
List of Figures x  
Glossary xiv  

## Chapter 1:

**Introduction** ................................. 1  
1.1 Registration ...................................... 3  
1.2 Registration Accuracy .......................... 5  
1.3 Contributions and Organization ............... 7  

## Chapter 2:

**Rigid-body Registration** ...................... 10  
2.1 Pairwise Registration ........................ 10  
2.2 Multi-body Registration ....................... 20  

iv
List of Tables

4.1 Performance comparison among UKF, Umeyama, and Horn algorithms when FLE is isotropic and identical. ........................................ 55
4.2 Performance comparison between UKF and Umeyama algorithms when FLE is inhomogeneous and anisotropic. ......................... 56
4.3 Performance comparison between UKF and Umeyama algorithm when FLE is anisotropic and identical. ................................. 59
4.4 Performance comparison between UKF and Umeyama algorithm when FLE is anisotropic and inhomogeneous. ......................... 60
4.5 Sensitivity of UKF registration algorithm to variance of anisotropic and identical FLE. ...................................................... 61
4.6 Sensitivity of UKF registration algorithm to variance of inhomogeneous and anisotropic FLE. ................................................ 62
4.7 Performance of UKF registration algorithm with ±20% error in variance of FLE. .............................................................. 64
4.8 Performance of UKF registration algorithm with ±50% error in variance of FLE. .............................................................. 65
4.9 Sensitivity of UKF registration algorithm to number of observations. ................................................................. 67
4.10 Performance comparison between UKF and EKF registration algorithm when number of registering points is 30. ..................... 68
4.11 Performance comparison between UKF and EKF registration al-
grithm when number of registering points is 150. . . . . . . . . . . . . 69
4.12 Comparison of convergence rate between UKF and ICP registration
algorithms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
4.13 Sensitivity comparison of UKF and ICP registration algorithms to out-
liers. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
5.1 Performance comparison between UKF and Pennec multi-body regis-
tration algorithms. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
5.2 Convergence rate of UKF multi-body registration algorithm. . . . . 107
6.1 Performance comparison between proposed and Fitzpatrick and West
TRE algorithms in estimation of TRE distribution. . . . . . . . . . . 130
6.2 Performance comparison between proposed and Fitzpatrick and West
algorithms in estimation of mean-squared value of TRE. . . . . . . 130
6.3 Performance comparison between proposed and Fitzpatrick and West
algorithms when data sets are symmetric. . . . . . . . . . . . . . . . 133
6.4 Performance comparison between proposed and Fitzpatrick and West
algorithms when data sets are asymmetric. . . . . . . . . . . . . . . . 134
6.5 Performance of proposed algorithm in estimation of mean-squared value
of TRE for different data sets. . . . . . . . . . . . . . . . . . . . . . . 142
6.6 Performance of ML TRE estimator when FLE is isotropic and inho-
mogeneous. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
6.7 Performance of ML TRE estimator when FLE is isotropic and identical.164
6.8 Performance of ML TRE estimator when FLE is anisotropic and iden-
tical. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
6.9 Performance of ML TRE estimator when FLE is anisotropic and inhomogeneous. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167

6.10 Mean (mm) and variance (mm²) of error difference between the computed root mean squared values of TRE at different target locations. 168
List of Figures

1.1 Preoperative data set collected from the anatomy of a patient. . . . . 3
1.2 Intraoperative data set generated from the anatomy of a patient. . . 3
1.3 Performed registration between preoperative and intraoperative data sets. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.4 A color-map showing the accuracy of the performed registration. . . 7

2.1 Three fractures of a pelvic bone. . . . . . . . . . . . . . . . . . . . . . . 21
2.2 Pairwise registration of multiple data sets. . . . . . . . . . . . . . . . . 22

4.1 Flowchart of UKF point-based registration algorithm for isotropic FLE. 44
4.2 Flowchart of UKF point-based registration algorithm for anisotropic FLE. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
4.3 Performance of UKF point-based registration algorithm and Umeyama algorithm when FLE is isotropic. . . . . . . . . . . . . . . . 52
4.4 Estimation of transformation parameters using UKF registration algorithm. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
4.5 Performance of UKF registration algorithm when FLE is anisotropic . 58
4.6 Sensitivity of UKF registration algorithm point ordering . . . . . . . 63
4.7 Performance of UKF registration algorithm with one observation . . 66
4.8 Performance of UKF registration algorithm with i observation . . . 66
4.9 Flowchart of UKF surface-based registration algorithm.  
4.10 Performance comparison between UKF and ICP registration algorithms when FLE is isotropic.  
4.11 Time complexity comparison between UKF and ICP registration algorithms when FLE is isotropic.  
4.12 Distribution of number of points used by UKF registration algorithm when FLE is isotropic.  
4.13 Performance of incremental ICP registration algorithm.  
4.14 Performance comparison between UKF and ICP registration algorithms when FLE is anisotropic.  
4.15 Time complexity comparison between UKF and ICP registration algorithms when FLE is anisotropic.  
4.16 Distribution of number of points used in UKF registration algorithm when FLE is anisotropic.  
4.17 Registration of CT and US images of a pelvic bone.  
4.18 Performance comparison between UKF and ICP registration algorithms in alignment of CT and US images of a pelvic bone.  
4.19 CT mesh of a scaphoid bone.  
4.20 US image of a scaphoid bone.  
4.21 Three-dimensional US mesh of a scaphoid bone.  
4.22 Registration of CT and US images of a scaphoid bone.  
4.23 Performance comparison between UKF and ICP registration algorithms in alignment of CT and US images of a scaphoid bone.  
5.1 Performance of UKF multi-body, point-based registration.
6.17 Performance of different TRE estimators when FLE is anisotropic and inhomogeneous. .......................................................... 170

6.18 Performance of different TRE estimators when FLE is isotropic and identical. .............................................................. 170
Glossary

**anisotropic**  Having properties that differ according to the direction of measurement. *Page 6.*

**CT**  Computer tomography: a medical imaging method employing tomography to visualize structure of body. *Page 2.*

**EKF**  Extended Kalman filter: a nonlinear version of Kalman filter which linearizes the nonlinearity using the first order Taylor series expansion. *Page 14.*

**EM**  Expectation maximization: an algorithm which is used in statistics for finding maximum likelihood estimates of parameters in probabilistic models. *Page 19.*

**fiducials**  Some prominent markers mounted on a body or extracted from a body to perform registration. *Page 4.*

**fixed data set**  Intraoperative images taken from patients during operations. *Page 11.*

**FLE**  Fiducial localization error: the error in calculating fiducials or markers from medical data sets. *Page 11.*

GTLS  Generalized total least mean squares: a least squares data modeling technique in which observational errors on both input and output data sets are taken into account. Page 15.

heterogeneous  Completely different; to be inhomogeneous and anisotropic. Page 28.

ICP  Iterative closest point algorithm: a method to register two point sets or surfaces. Page 5.

IGS  Image-guided surgery: a real-time correlation of operative field to a preoperative imaging data set. Page 1.

inhomogeneous  Not similar and the same kind. Page 6.

isotropic  Identical in all directions; invariant with respect to direction. Page 6.

ML  Maximum likelihood: a statistical method used for estimation of some parameters or fitting some data to a model. Page 19.

moving data set  Preoperative images which are registered to intraoperative images. Page 11.

MRI  Magnetic resonance imaging: a medical imaging technique most commonly used in radiology to visualize structure and function of body. Page 2.

PCA  Principal component analysis: a vector space transform which is usually used for reducing multidimensional data sets to lower dimensions for analysis. Page 110.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRE</td>
<td>Target registration error: the root mean square distance between a pair of corresponding fiducial positions that is not used in registration. <em>Page 27.</em></td>
</tr>
<tr>
<td>UKF</td>
<td>Unscented Kalman filter: a nonlinear version of Kalman filter which uses unscented transform to deal with the nonlinearity. <em>Page 7.</em></td>
</tr>
<tr>
<td>US</td>
<td>Ultrasound: a cyclic sound pressure used to visualize structure of body. <em>Page 2.</em></td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Image-guided surgery (IGS) can be defined as a real-time correlation of the high-quality images taken preoperatively to the operative images obtained from patients during a surgery. This technology was first developed to give neurosurgeons a more reliable and controllable view of where healthy tissue ends and brain abnormality begins. However, adaptation of this technique to other surgeries such as orthopedic and cranio-facial surgeries rapidly became the leading indication of this technology [106].

Because of the quality and accuracy that the IGS systems provide, surgeons can create an optimal and detailed preoperative plan for surgery. For instance, in orthopedic surgery, the surgeons can manipulate the preoperative images to pinpoint the optimal location for a pedicle screw on a spine or a knee to maximize the patients’ mobility. IGS systems not only help surgeons preoperatively to obtain a better view of their patients’ conditions and problems, but they also offer benefits during operations. Intraoperatively, surgeons can use IGS to change the view of the anatomy in real-time. This constant flow of information helps them to make minute adjustments
to ensure they are operating on the exact areas which need to be treated.

Besides aiding surgeons, IGS benefits patients undergoing surgeries. It can be effectively used to reduce the size of patient’s incisions, and hence, can make operations less invasive. It also helps the surgeon to shorten the operation time. Because of these properties, IGS provides new solutions for patients with multiple medical problems, patients who may not be able to endure a long and invasive surgery, or patients whose medical problems previously precluded surgery.

Today, IGS normally begins with obtaining a set of medical images from patients’ anatomy. Computer Tomography (CT), Ultrasound (US) or Magnetic Resonance Imaging (MRI) are usually used to generate medical images from patients. These images which are taken preoperatively, are utilized to plan the surgery. The plan and images are then transferred to the operational room, and, before the operation, the images are brought up on the screen of IGS systems to finalize the preplanned operation, as shown in Figure 1.1. Here, a significant problem arises because the patients’ anatomy may have been moved after taking the preoperative images and before starting the operation. Since the position of the patient’s anatomy has changed, the surgical plan made by surgeons based on the preoperative images is no longer valid. Therefore, a mapping is crucial in order to update the plan by aligning (shifting) the preoperative images to the actual patient’s anatomy in the operation room which is identified by taking intraoperative images from the patient, as shown in Figure 1.2. This mapping is called registration.

Registration generates a mapping between the position of the patient anatomy in the surgical field and the corresponding location in the preoperative images. It is an important step in IGS, since inaccurate and unreliable registration may lead surgeons
to operate on the wrong areas of patients. Hence, a reliable image registration technique is imperative in IGS systems. For example, Figure 1.3 displays a registration result between CT mesh data generated preoperatively from a pelvic cadaver bone, and points collected intraoperatively from the pelvic cadaver in the operating room. In this case accurate registration would lead to a successful osteotomy operation [1].

As part of registration, surgeons also require a measure that computes the accuracy of the performed registration to help them decide whether to accept or reject the registration before continuing with the planned surgery.

![Figure 1.1: Preoperative data set loaded on the IGS system to be used in operation [6].](image1)

![Figure 1.2: Collecting intraoperative data set from the patient anatomy to register them with the preoperative images [Courtesy of Dr. Gabor Fichtinger at Queen’s University].](image2)

1.1 Registration

Registration is normally split into two main categories: nonrigid and rigid. Nonrigid registration algorithms align images taken from a deformable object. On the other hand, when an object or anatomy is rigid, such as bone, then a rigid registration
can be used to align the preoperative and intraoperative images. Rigid registration is defined as a function which maps preoperative and intraoperative images by a rotation matrix and a translation vector. The latter registration has applications in many clinical image-guided surgeries such as orthopaedic, hip, spine and cranio-facial surgeries. If registration happens between two data sets, then it is called pairwise, and if more than two data sets are aligned at the same time, then the registration is called multi-body or groupwise. The literature contains many pairwise rigid registration algorithms which align the preoperative and intraoperative data sets from a rigid object. These algorithms are usually based on features and are called feature- or point-based registration. Features are either extracted from the data sets or created artificially by mounting markers, called fiducials, on the patient’s anatomy in the surgical site. Due to the feature localization error, the performed registration is not usually very accurate. To improve the accuracy of the registration, the number of...
features between the data sets should be increased. But, in real applications, it is sometimes impossible to extract or locate many features from the patient’s anatomy to perform an accurate registration. To solve this problem, surface-based registration algorithms have been introduced in the literature. The Iterative Closest Point (ICP) algorithm [11] and its variants are the most widely used methods to improve the registration results in IGS systems. To do so, these methods choose the closest points between the two data sets and assume they are reliable corresponding points.

If the initial feature-based registration is not sufficiently accurate, then the data sets are not preliminarily closely aligned. Therefore, this misalignment means that the closest points algorithm does not return reliable correspondences between the data sets. That incorrect correspondences may lead the ICP algorithm to inaccurately register the data sets means that the ICP algorithm is sensitive to the initial alignment error.

Hence, a robust rigid body registration algorithm is highly desirable in IGS systems where rigid-body registrations play an important role. The first part of this dissertation describes our attempts to derive a new robust rigid registration algorithm which is less sensitive to the misalignment error and noise in data sets.

1.2 Registration Accuracy

The performed registration is usually imperfect due to the noise (such as fiducial localization error) in data sets. So, a measure is required that reports the accuracy of the performed registration. For instance, Figure 1.4 displays the accuracy of the performed registration in Figure 1.3. Surgeons need this information to decide if the performed registration is sufficiently accurate to continue the image-guided surgery.
This measure can also give surgeons a reliable insight into how the registration error propagates to regions where they do not have access.

Registration accuracy is usually estimated by performing computer simulations and experiments with phantoms and cadavers where the ground truth is known. But, Sibson [113], in 1979, was the first who analytically derived an expression for registration error. Following this work, Fitzpatrick and West [36] derived a novel closed-form solution that estimated the mean-squared value of registration error at a target location. They also used Monte-Carlo simulation to compute the distribution of the registration error at that target. In their derivations, they assumed that the noise perturbing the data sets had an isotropic and identical zero-mean Gaussian distribution.

In real clinical applications, noise in the data sets may not have an isotropic and identical Gaussian distribution. Due to the properties of the medical instruments used for collecting preoperative and intraoperative data sets, noise perturbing the data sets usually has anisotropic and inhomogeneous distribution. For instance, noise in CT and MRI images has an anisotropic and inhomogeneous distribution, due to the lower out of plane resolution of images versus the inplane image resolution. Similarly, in ultrasound images, the variable depth-dependent elevation beam profile would introduce a nonuniform uncertainty to the measurements from these images. Therefore, we require an algorithm that estimates the characteristics of the registration error when noise in the data set has an anisotropic and inhomogeneous distribution. Anisotropic and inhomogeneous distribution respectively means that the distribution is different along each orthogonal axes and at each point in the data set.

The second part of this dissertation describes our attempt to derive a solution to
estimate the distribution of the registration error at a target location when noise in the data sets has a general type of distribution.

![Figure 1.4: Estimation of the registration accuracy, after performing registration between preoperative and intraoperative data set.](image)

**1.3 Contributions and Organization**

This dissertation makes two main contributions:

1) We introduce a robust and accurate pairwise point-based registration algorithm in the presence of arbitrary noise in the data sets. Since the Unscented Kalman Filter (UKF) can appropriately account for noise in a system, and optimally estimate desired parameters, it is used in the proposed registration algorithm to compute the registration parameters. We then extend the proposed pairwise point-based registration algorithm to a new pairwise surface-based registration method. Due to the
properties of UKF, the new algorithm is more robust and accurate than the well-known ICP registration method. Finally, the proposed pairwise registration methods are expanded to the multi-body point and surface-based registration algorithms, respectively.

2) We show that the output of the proposed registration algorithms can be employed to estimate the accuracy of the performed registration, i.e., the distribution of the registration error. We derive the first closed-form solution that estimates the distribution of registration error at any desired target location when the noise perturbing the data sets has an arbitrary Gaussian distribution. We also mathematically derive expressions for estimating the mean-squared value of the registration error at a target, and the loci of the points having the same mean-squared value of the registration error. These expressions, respectively, indicate that the mean-squared value of the registration error depends on the geometry of the data sets, and that the loci of the points with the same registration error are ellipsoids. Finally, we prove that, for special cases where noise has isotropic and identical Gaussian distribution, the proposed derivations simplify to those previously derived in the literature.

The rest of this dissertation is organized as follows:

Chapter 2 briefly reviews point and surface-based registration algorithms when two or more data sets should be aligned together. Then, accuracy estimation of the performed registration is studied.

Chapter 3 provides background material for the UKF algorithm. This algorithm will be used in the proposed point and surfaced-based registration algorithms in Chapters 4 and 5, respectively. We also use this algorithm in Chapter 6 to estimate the
distribution of the registration error.

**Chapter 4** describes and analyzes new pairwise point- and surface-based registration algorithms based on UKF. We show that the proposed algorithms accurately register two data sets, and are less sensitive to the noise and initial alignment error.

**Chapter 5** describes and analyzes new multi-body point- and surface-based registration algorithms based on UKF. We show that the proposed algorithms can simultaneously and accurately register multiple overlapping data sets.

**Chapter 6** describes and analyzes novel derivations that estimate the accuracy of the performed registration at any desired location. We show that the proposed formulations accurately estimate the distribution and the mean-squared value of registration error when noise perturbing the data sets has an arbitrary distribution.

**Chapter 7** summarizes the significant contributions of this dissertation.

Note: This work has been presented in the *IEEE Transactions on Medical Imaging* [84, 87], and the proceedings of the following conferences: *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* [82, 85], *International Conference of the Medical Imaging (SPIE)* [86, 89, 88], and *International Conference of the Medical Image Computing and Computer Assisted Intervention (MICCAI)* [71, 81, 83, 90].
Chapter 2

Rigid-body Registration

In this chapter, we define pair and groupwise registration problems, and introduce the most relevant of those registration algorithms to this dissertation. For further information on diverse registrations algorithms used in medical imaging, refer to the review articles by Audette et al. [67], Lavallee [62], and Maintz and Viergever [72].

2.1 Pairwise Registration

Pairwise registration of two three-dimensional data sets is an important problem in medical imaging and computer vision, where two data sets must be aligned in a common reference frame. One application of such registration is in computer-assisted and image-guided surgery where surgeons wish to register two sets of images, taken preoperatively and intraoperatively from a patient in order to deduce surgical guidance information, and to correct for patient movement. In what follows, we first explain pairwise point-based registration, and then, we discuss pairwise surface-based registration.
2.1.1 Pairwise Point-based Registration

The pairwise point-based rigid body registration problem can be defined as finding the optimum transformation that aligns two point sets, given a set of corresponding points in two different coordinate frames. Sometimes, this is also called the absolute orientation problem.

Many clinical applications, such as computer-assisted orthopaedic surgeries, use the points (fiducials) that are identifiable in a preoperative imaging modality and physically measurable during surgery to perform the registration, and therefore, to transfer the preoperatively selected points (moving data set) to those collected intraoperatively (fixed data set). In the most commercial computer-assisted and image-guided surgery systems, this transformation is rigid and consists solely of a rotation matrix and a translation vector.

To mathematically define the problem, we represent the moving and fixed data sets by $\mathbf{U}$ and $\mathbf{Y}$, respectively. $\mathbf{U}$ and $\mathbf{Y}$ are each a $3 \times N$ matrix whose columns correspond to the location of the points in a three-dimensional space ($N$ indicates the number of points in each data set). Each $i$th column in the moving and fixed data sets represents a pair of corresponding points in the two data sets. Since the preoperative and intraoperative localized points or fiducials are imprecise, it is assumed that the $i$th pair of points is perturbed by fiducial localization error (FLE) as follows:

\begin{align}
\mathbf{y}_i &= \mathbf{\hat{y}}_i + \mathbf{n}^y_i, \\
\mathbf{u}_i &= \mathbf{\hat{u}}_i + \mathbf{n}^u_i,
\end{align}

where $\mathbf{\hat{y}}_i$ and $\mathbf{\hat{u}}_i$ are the noiseless points in the fixed and moving data sets; $\mathbf{n}^u_i$ and $\mathbf{n}^y_i$ are assumed to be two independent, zero-mean Gaussian random variables with general covariance matrices $\Sigma^u_i$ and $\Sigma^y_i$, respectively, that model FLE. For instance, in
the case of isotropic zero-mean Gaussian noise, where FLE has the same distribution along each orthogonal axes, \( \Sigma^u_i \) and \( \Sigma^y_i \), can be simplified as \( \sigma^2_{i,u} I_{3 \times 3} \) and \( \sigma^2_{i,y} I_{3 \times 3} \), respectively.

The most common way to express the registration problem is to determine a \( 3 \times 3 \) rotation matrix \( R \) and a \( 3 \times 1 \) translation vector \( t \) such that the following cost function is minimized

\[
C(R, t) = \frac{1}{N} \sum_{i=1}^{N} (y_i - [R u_i + t])^T (\Sigma^y_i)^{-1} (y_i - [R u_i + t]),
\]

where \( \Sigma^y_i \) is the covariance matrix of the FLE for the \( i \)th pair of corresponding points in the moving and fixed data sets. \( \Sigma^y_i \) can be simply computed from \( \Sigma^u_i \) and \( \Sigma^y_i \) as

\[
\Sigma^y_i = \Sigma^y_i + R \Sigma^u_i R^T.
\]

If \( \Sigma^u_i = \sigma^2_{i,u} I_{3 \times 3} \) and \( \Sigma^y_i = \sigma^2_{i,y} I_{3 \times 3} \) (FLE is isotropic in the two data sets), then (2.4) can be simplified as

\[
\Sigma^y_i = (\sigma^2_{i,u} + \sigma^2_{i,y}) I_{3 \times 3}.
\]

By substituting (2.5) into (2.3), (2.3) can be written as

\[
C(R, t) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sigma_i^2} (y_i - [R u_i + t])^T (y_i - [R u_i + t]).
\]

If FLE in the two data sets is isotropic and its distribution is also the same for each pair of corresponding points in the data sets, \( i.e., \Sigma_i = \sigma^2 I_{3 \times 3} \), then (2.6) can be further simplified as

\[
C(R, t) = \frac{1}{Ns^2} \sum_{i=1}^{N} (y_i - [R u_i + t])^T (y_i - [R u_i + t]) = \frac{1}{s^2} E[\|y_i - (R u_i + t)\|],
\]

where \( E \) and \( \| \| \) are the expected value and norm 2 operators, respectively.

The absolute orientation problem—finding the optimum \( R \) and \( t \) that minimize
has been shown to have closed-form solutions when FLE has an isotropic zero-mean Gaussian distribution. Different approaches have been used in the literature to analytically minimize (2.7), and compute rotation and translation parameters.

Farrell and Stuelpnagel [33], and Schönemann [108], at the same time, published the first solution in 1966 which was based on orthogonal Procrustes to minimize (2.7). Their result was rediscovered independently by Golub and van Loan [42] in 1983. Arun et al. [2] derived a closed-form solution to minimize (2.7) based on the singular value decomposition (SVD) in 1987. Horn et al. proposed more closed-form solutions for the absolute orientation problem in 1987 [47] and 1988 [48], respectively. The latter algorithms, which are similar to Arun’s, use a unit Quaternion vector [47] and an orthogonal matrix [48] to estimate the rotation matrix \( R \). Similarly, Walker et al. [127], in 1991, devised another answer to the absolute orientation problem based on dual Quaternions. In 1995, Lorusso et al. [64] showed that these closed-form solutions are almost the same, and that no superior method exists for all cases.

Umeyama [126] argued that Horn and Arun algorithms sometimes failed to give the correct rotation matrix when data sets are excessively noisy. Therefore, Umeyama proposed a strict solution to minimize (2.7) based on the Lagrange Multipliers. Kan
tani [56] proposed a simplified version of the Umeyama algorithm that only estimated the rotation matrix \( R \) between the two data sets. Wang et al. [129] presented another closed-form solution for the absolute orientation problem based on the linear sub-space method. In order to minimize (2.7), this algorithm first computes the translation parameters, and then independently calculates the rotation parameters. The rotation parameters are calculated in a similar manner to that of Horn’s. Maurrer et al. [77] presented another closed-form solution to estimate the rotation and
translation which was similar to that proposed by Arun.

All the algorithms cited above assume that FLE perturbing the registering data sets has an isotropic zero-mean Gaussian noise distribution. However, in real applications, the distribution of FLE may not be isotropic or even Gaussian. In what follows, registration algorithms dealing with anisotropic Gaussian noise in the registering data sets are discussed. When FLE has an anisotropic zero-mean Gaussian distribution, (2.3) should be minimized to compute the rotation and translation parameters. Equation (2.3) has not been yet analytically minimized, but many algorithms in the literature iteratively minimize (2.3) to calculate $\mathbf{R}$ and $\mathbf{t}$.

In 1997, Pennec and Thirion [98] proposed an iterative solution to minimize (2.3). They employed the Extended Kalman Filter (EKF) [130] algorithm to iteratively estimate the transformation parameters in the presence of isotropic or anisotropic Gaussian noise. Ma et al. [70] proposed the Unscented Particle Filter (UPF) algorithm [78] to estimate the transformation parameters by minimizing (2.3). The UPF algorithm, with its high computational complexity, is a powerful technique that can be used to estimate unknown parameters from nonlinear functions perturbed by any type of noise distribution [79]. Koschat et al. [59] suggested an iterative algorithm, based on the weighted Procrustes problem, to register two pairs of data sets when the points are perturbed by anisotropic noise. Assuming there is no translation between the data sets, their algorithm, similar to Arun’s, uses SVD to iteratively determine the rotation matrix $\mathbf{R}$ by minimizing (2.3). Chu et al. [17] utilized the steepest descent algorithm to minimize (2.3), and to find the transformation parameters. Batchelor et al. [4] compared these two iterative algorithms using numerical simulations, and showed that the performance of both algorithms is approximately the same. Ohta et
al. [93] developed another iterative algorithm to register two sets of correspondences in the presence of anisotropic noise. Their algorithm iteratively minimizes (2.3) using the re-normalization technique [57]. Ohta and Kanatani [94] introduced a GTLS solution [107] to fit a rotation matrix between two three-dimensional data sets in the presence of anisotropic and inhomogeneous noise. Finally, in 2005, Balachandran et al. [3] proposed another algorithm for registration of two sets of corresponding points in the presence of anisotropic additive Gaussian noise. Their algorithm first pre-estimates the rotation matrix $\mathbf{R}$ based on the assumption that FLE is isotropic. Then, the anisotropic noise is incorporated in the calculation of the centroids of the two data sets to re-estimate the rotation matrix $\mathbf{R}$.

In this dissertation, we derive a new registration algorithm to register two data sets when FLE has an arbitrary zero-mean Gaussian distribution. The proposed algorithm which is based on an estimator called UKF [54], sequentially minimizes (2.3) to estimate the rotation and translation parameters. This algorithm is much less computationally burdensome than the UPF registration technique proposed by Ma et al. [70] and thus can be applied to large data sets.\textsuperscript{1} Also, it is more accurate than the algorithm proposed by Pennec [98], since the UKF algorithm has been shown to be more accurate than EKF [128]. Another benefit of the proposed algorithm is that, in contrast to the previous art, it can compute the transformation parameters along with their variances. The variances can be further used to estimate the accuracy of the performed registration. Details of the proposed algorithm are presented in Chapter 4.

\textsuperscript{1}The computational complexity of UPF is as $m$ times as the computational complexity of UKF, where $m$ is the number of particles in UPF algorithm.
2.1.2 Pairwise Surface-based Registration

In the pairwise surface-based rigid body registration, points which are physically and intraoperatively measured from the surface of a patient anatomy are matched with the model of the surface that are preoperatively generated. In computer-assisted and image-guided orthopedic surgery, the model is usually computed from a three-dimensional imaging modality such as MRI, CT and ultrasound. The measured points are obtained using a tracking system which records the three-dimensional location of a calibrated probe such as an ultrasound transducer and a stylus pointer [68].

The goal of surface-based registration is to find optimal \( R \) and \( t \) that minimize the distance between a set of measured points \( U_{3 \times N} \) and the model \( Y_{3 \times M} \). In surface-based registration, the number of points in \( U \) is usually less than \( Y \), i.e., \( N \leq M \).

To estimate the registration parameters, first, Besl and McKay [11], and then Chen and Medioni [16] popularized a function that is the mean-squared error among the registered point sets \( U \) and their closest points in \( Y \), as

\[
C(R, t) = \frac{1}{N \sigma^2} \sum_{i=1}^{N} (f_{\text{ClP}}(Y, u_i) - [Ru_i + t])^{T} (f_{\text{ClP}}(Y, u_i) - [Ru_i + t]),
\]

where \( f_{\text{ClP}} \) is the operator that finds the closest point in the data set \( Y \) to \( u_i \). The authors proposed a novel method, called Iterative Closest Point (ICP) algorithm, to estimate the optimal \( R \) and \( t \) that minimize (2.8). There are two main steps in the ICP registration algorithm: the first is the matching of points between surfaces (finding a set of correspondences), while the second is the calculation of the optimal rigid transformation mapping the correspondences to each other. The algorithm begins with the assumption that the surfaces are preliminarily aligned such that the closest points between the two surfaces are reliable candidates of true correspondences. Then each point in the moving data set is matched with the closest point in the fixed
surface to generate a set of correspondences. Finally, an optimal transformation that aligns the corresponding points is computed by minimizing (2.8) and using Arun algorithm [2]. This transformation is then applied to the moving surface and the procedure is repeated till convergence is achieved.

The advantages of the ICP algorithm are that it is simple to implement, and has fast execution time. One of the main drawbacks of the ICP algorithm is its sensitivity to the initial alignment. If the two surfaces are not initially closely aligned, then the closest points between the two surfaces might result in false matches. Subsequently, the false matches may cause the ICP algorithm to calculate an inaccurate transformation which results in failure of the ICP algorithm to find the optimal solution. The below authors have proposed modifications to the matching step to account for outliers, and to reduce the number of false matches.

Masuda et al. [73] developed a robust method to perform surface-based registration in the presence of outliers. Their method is an integration of the ICP algorithm with a random sampling procedure to find the closest points, and the least median of squares estimator to find the best transformation reported by the ICP algorithm. Trucco et al. [124] presented a similar algorithm which also used the least median square estimator to withstand the outliers.

Zhang et al. [140], Schutz et al. [109], Feldmar et al. [34] and Turk et al. [125] all employed thresholding to remove outliers from the correspondences. In those algorithms, corresponding points having a distance greater than a certain threshold are assumed to be outliers, and are not considered when updating the transformation parameters. Ma et al. [69] identified some problems and limitations with the thresholding approach and presented a robust version of the ICP algorithm for image-guided
and computer-assisted orthopedic surgery. In their algorithm, outliers are detected and handled by integrating a robust M-estimator that estimates the transformation parameters with the ICP algorithm. Blais and Levine [13] used an algorithm called very fast simulated re-annealing [50] to estimate the transformation parameters in the ICP algorithm. Although the algorithm is able to traverse the local minima, it is extremely slow in converging to the global minima when many shallow local minima have surrounded the global solution. To speed up the surface-based registration algorithm, Luck [65] introduced a hybrid surface registration algorithm by combining the ICP algorithm with the simulated annealing [58] method. Maurer et al. [75, 76] and Doari et al. [28] used the weighted least-squares algorithm to robustly estimate the transformation parameters in the ICP algorithm. Their algorithm is fast, but less robust than the simulated annealing registration algorithm [13]. Fitzgibbon [35] and Champleboux et al. [14] modified the ICP algorithm by employing the Levenberg-Marquardt [101] method to estimate the transformation parameters by minimizing (2.8). Although the Levenberg-Marquardt algorithm is more robust than the weighted least-mean-squares algorithm, it is slower to find the optimum registration solution.

Silva et al. [116], Robertson and Fisher [104] and Yamany et al. [139] used the Genetic algorithm to estimate the transformation parameters in the ICP algorithm. They showed that their algorithm can finally converge to the optimum registration parameters but it may take a long time.

Betting et al. [12], Pennec and Thirion [98], and Stoddart et al. [120] presented modified versions of the ICP algorithm that use the EKF algorithm to estimate the transformation parameters as well as their variances. Ma and Ellis [70] proposed the UPF algorithm as a better optimizer than EKF to align the corresponding points
obtained from the closest point algorithm. Their algorithm is more robust than EKF registration method, but its time complexity is greater. Esteper et al. [31] investigated the use of a GTLS [93] approach rather than a least-squares algorithm for estimating the transformation parameters in the ICP algorithm. They showed that their algorithm is more robust than the standard ICP algorithm for dealing with outliers.

Finally, Grenger et al. [43], Rangarajan et al. [103], Chui and Rangarajan [19], Luo et al. [66], Wells [131] and Dellaert [25] introduced variants of the ICP registration algorithm based on Expectation Maximization (EM) algorithm [91]. Specifically, they formulated the ICP algorithm as a Maximum Likelihood (ML) estimation of the transformation and the corresponding points. By considering the corresponding points as a random matrix with a known probability distribution, the ML estimation of the transformation parameters and the corresponding points generates a new cost function which is effectively solved using the EM algorithm [26].

In this dissertation, we would like to combine the standard ICP and UKF algorithms to propose a new surface-based registration method. In the proposed approach, UKF can be used as a weighted least-mean-squares algorithm to estimate the registration parameters by iteratively minimizing the following cost function

\[
C_{(R,t)} = \frac{1}{N} \sum_{i=1}^{N} \left( f_{\text{CLP}}(Y, u_i) - [Ru_i + t] \right)^T \left( \Sigma_i^u + R \Sigma_i^a R^T \right)^{-1} \left( f_{\text{CLP}}(Y, u_i) - [Ru_i + t] \right),
\]

(2.9)

Compared to the prior art, the proposed algorithm can easily account for any type of Gaussian noise in the registering data sets, and can also assign different weights to each of the computed corresponding points. These two advantages make the proposed algorithm more robust than the standard ICP algorithm to the outliers, initial
alignment error, and the noise in the registering data sets. Furthermore, since the proposed algorithm, in contrast to the prior art, is a sequential registration algorithm, it makes real-time registration more feasible in real clinical applications.

2.2 Multi-body Registration

In certain clinical applications, more than two anatomical pieces need to be registered in a common reference frame. For instance, in computer-assisted and image-guided orthopedic trauma surgery, surgeons need to simultaneously reassemble and return the pieces of a fractured bone back to their original correct positions. In this scenario, there may be more than two bone fractures, and therefore, a multi-body registration algorithm is required to simultaneously register all the bone fractures together.

In what follows, we first explain the multi-body, point-based registration algorithms where there are sets of corresponding points in different coordinate frames that should be simultaneously registered in a common coordinate frame. Then, we extend the discussion to the multi-body surface-based registration where there are overlapping surfaces that should be registered together.

2.2.1 Multi-body Point-based Registration

In multi-body registration, at least three overlapping areas (data sets) should be aligned to each other in contrast to simple pairwise registration where there are only two data sets to be registered. See, in Figure 2.1, three pelvis bone fractures that we would like to globally register back to their correct healing positions. This can be effected by estimating the optimal transformations $T_1$, $T_2$, $T_3$, $T_1^2$, and $T_3^2$ at the
same time. It means that all the overlapping areas (among all the data sets) should be simultaneously considered in a registration algorithm to estimate the transformation parameters.

Figure 2.1: Multiple data set registration of three pelvis bone fractures to the whole bone model (template). $T_1$, $T_2$, $T_3$, $^2T_1$ and $^2T_3$ are the rigid transformations mapping fracture 1 to the bone template, fracture 2 to the bone template, fracture 3 to the bone template, fracture 1 to fracture 2, and fractures 3 to fracture 2, respectively.

One may argue that pairwise registration algorithms may also be utilized to solve the multi-body registration problem for multiple data sets by sequentially registering each two pairs of the data sets. However, as shown in Figure 2.2, this may lead to an uneven spread of registration error among the data sets [112], and therefore, a groupwise (multi-body) registration algorithm is needed to evenly distribute the
registration error among all the overlapping regions.

Figure 2.2: Uneven distribution of error over all overlapping areas caused by sequentially using pairwise registration algorithms.

To mathematically define the global registration problem based on the known correspondences, let’s assume there are $M$ data sets ($S^1, S^2, \ldots, S^M$) that may overlap with each other. The problem consists of determining $M$ rigid transformations ($T_1, T_2, \ldots, T_M$) such that the cost function $C$ is minimized,

$$C(T_1, \ldots, T_M) = \sum_{\alpha=1}^{M-1} \sum_{\beta=\alpha+1}^{M} \sum_{k=1}^{N_{\alpha,\beta}} (T_{\alpha} p_{\alpha,\beta}^k - T_{\beta} p_{\beta,\alpha}^k)^T [\Sigma_{k}^{\alpha,\beta}]^{-1} (T_{\alpha} p_{\alpha,\beta}^k - T_{\beta} p_{\beta,\alpha}^k),$$  \hspace{1cm} (2.10)

where the weight matrix $\Sigma_{k}^{\alpha,\beta}$ is given and models the covariance of the fiducial localization error (FLE) in $p_{\alpha,\beta}^k$ and $p_{\beta,\alpha}^k$, where $p_{\alpha,\beta}^k$ and $p_{\beta,\alpha}^k$ are the corresponding points between $S^{\alpha}$ and $S^{\beta}$ data sets, respectively, and $N_{\alpha,\beta}$ is the number of corresponding points between the two data sets.

It should be noted that this problem is degenerate since, if the same transformation applies to all data sets, it does not alter the registration. This degeneracy can be removed by assuming that one of the transformations, such as $T_1$, to be the identity. Therefore, there are $M - 1$ transformations remaining to be estimated.
CHAPTER 2. RIGID-BODY REGISTRATION

In the past, many multi-body point-based registration algorithms were proposed to solve the multiple data set registration problem. Stoddart and Hilton [119] used an iterative numerical solution based on the gradient descent algorithm to minimize (2.10), and to simultaneously derive the transformation parameters. Their algorithm assumes that the data sets are perturbed by isotropic noise, i.e., \( \Sigma_k^{\alpha,\beta} = \sigma_k^2 I_{3\times3} \). At the same time, Benjemma and Schmitt [8] and Williams et al. [135] presented similar iterative solutions to minimize (2.10) and to estimate the transformations parameters when \( \Sigma_k = I_{3\times3} \). Their algorithm is a generalized and iterative formulation of the Horn algorithm [47]. In each iteration, one data set is assumed to be in the reference frame and the other data sets to be simultaneously registered to it. This iterative procedure continues till the algorithm converges to the nearest local minima. Pennec [97], in the presence of isotropic noise in the data sets, used the concept of “mean shape” to iteratively minimize (2.10). Cunnington and Stoddart [23] compared the above mentioned three global registration algorithms. They concluded that the algorithms’ performances are almost comparable but that Pennec algorithm is easiest to implement.

Goldberger [41] used EM algorithm to globally register multiple d-dimensional data sets perturbed by isotropic noise. Williams and Bennamoun [134] proposed a novel weighted least-squares algorithm to solve the multiple registration problem in the presence of anisotropic noise. By utilizing the Lagrange Multiplier and an iterative numerical technique (Newton method) they solved the transformations parameters simultaneously by minimizing (2.10). One of the main advantages of their algorithm is that it can also compute the variance of the estimated transformation parameters. The same authors [136] similarly described an iterative algorithm to minimize
(2.10) using the classic Gauss method. Pottmann et al. [100] developed an iterative algorithm to simultaneously register multiple data sets, based on the first-order kinematical analysis. To estimate the transformation parameters, they minimized (2.10) when $\Sigma_k^{x,\beta} = I_{3\times3}$. Finally, Krishnan et al. [60] used the manifold concept to reformulate (2.10), and used an iterative scheme based on the Newton’s method to minimize the new cost function and estimate the transformation parameters.

All of the mentioned algorithms iteratively minimize the cost function $C$ by considering all corresponding points at the same time. Instead, we adopt an iterative minimization approach, using the UKF algorithm, that simultaneously estimates the transformation parameters by incrementally and sequentially feeding the corresponding points to the registration algorithm. Since the optimization method usually converges well before all the corresponding points are used in the registration process, the algorithm is computationally efficient and can potentially make real-time clinical applications feasible. Furthermore, the proposed approach can also compute the variance of the estimated transformation parameters, which, in turn, can be used to estimate the accuracy of the performed registration.

### 2.2.2 Multi-body Surface-based Registration

In addition to multi-body point-based registration approaches, some research in the literature performs groupwise surface-based registration. In the latter, the algorithms assume that the correspondences among surfaces are not known, but that the surfaces are roughly aligned such that the closest points among the surfaces reliably estimate the correct correspondences.

Eggert et al. [29], Gagnon et al. [40], Bergevin et al. [10], Benjemaa and Schmitt
[7], Pulli [102], Blais and Levine [13], and Nishino and Ikeuchi [92] assumed that the surfaces are roughly aligned, and iteratively used the ICP algorithm to finely register the surfaces. One surface is assumed to be in the reference frame, and, in each iteration, $M - 1$ transformations based on the ICP registration algorithm for the other surfaces are estimated. This procedure is repeated till the algorithm converges to the closest local minima.

Kamgar-Parsi et al. [96] modelled the multiple surface registration problem with a mechanical system involving many springs between pairs of surfaces. The strength of the spring represents the amount of overlap between the surfaces. Their solution to the multiple surface registration (estimating the transformation parameters) is then defined as the equilibrium state of the mechanical system. Then, they used an iterative optimization algorithm to find the equilibrium state. Sharp and Sang [112] proposed another technique, using pairwise registration as a reliable initial alignment, to perform multiple surface registration. First, they performed pairwise registration between neighboring surfaces and then they evenly distributed the accumulation error over all registered overlapping areas. This procedure was repeated until the algorithm converged to a nearest local minimum. Dorai et al. [27] and Guehring et al. [44] also used pairwise registration techniques to preliminary align the surfaces. Then, the ICP algorithm was sequentially utilized among each overlapping region to simultaneously register all the surfaces together. This procedure was repeated till the change in the transformation parameters would fall below a preset threshold. Finally, Silva et al. [115] proposed genetic algorithms to perform groupwise registration among a set of surfaces.

Most of the cited algorithms are based on the ICP algorithm that is sensitive to
the initial alignment error and noise (FLE) in data sets. Furthermore, they assume that noise in the data sets has identical distribution. In this dissertation, we will use the UKF algorithm to register multiple surfaces in the presence of FLE that might have an inhomogeneous Gaussian distribution. Since UKF is a weighted least-mean-squares algorithm, it can easily deal with noise inhomogeneity in the data sets.

### 2.3 Accuracy of the Performed Registration

The question of how to gauge the accuracy of the performed registrations between preoperative and intraoperative data sets is crucial for computer-assisted and image-guided surgeries. Accuracy can be used as a measure to accept or repeat the registration, and surgeons can use this measure to gain an understanding of the registration accuracy at their desired locations within a patient anatomy. If the accuracy of the registration is low at their desired locations, they can then repeat the registration. The question, “How can the accuracy of the performed registration be measured or what is the relationship between the accuracy of registration and noise (FLE) in the registering data sets?”, are discussed in this section.

Maurer et al. [75, 76] studied the performed registration error for point-based registration by using computer simulations. In their work, to analyze the accuracy of the performed registration, they proposed three error measures as follows:

1) **Fiducial (point) localization error (FLE):** the error in extracting the points from data sets.

2) **Fiducial (point) registration error (FRE):** the root mean-squared distance among corresponding point pairs after registration.
3) Target registration error (TRE): the distance after registration between a corresponding point pair which was not used in the registration process.

Using numerical simulations the authors verified that there is a mathematical relation among FLE, FRE and TRE.

Likewise, the FLE, FRE and TRE properties have been studied for a long period of time. The distribution of FRE in terms of FLE was first discovered by Sibson in 1979 [113]. Hill et al. [45], Evans et al. [32], and Darabi et al. [24] also used numerical simulations to examine FRE and TRE properties in terms of FLE. Their findings numerically showed that TRE was inversely proportional to the square root of the number of fiducials (points) used in the registration. Maurer et al. [76] also showed that TRE was proportional to FLE, inversely proportional to the square root of the number of the points (fiducials) used in the registration, and strongly dependent on the configuration of the points. Ellis et al. [30] proposed a new method to evaluate fiducial registration accuracy in the absence of ground truth knowledge of the fiducials' locations. They showed that the configuration of the fiducials had a strong effect on the registration error. By using the Monte-Carlo simulation, they illustrated that TRE depends on the distance between the target and the centroid of the fiducials used in the registration.

Fitzpatrick et al. [37, 38] first analytically derived an approximation of the root mean-squared value of TRE in 1998. Their results demonstrated that mean-squared value of TRE is proportional to the mean-squared value of FLE and inversely proportional to the square root of the number of fiducials used in the registration. In 1999, West et al. [132] presented an approximation to the TRE distribution by ignoring an anisotropy in the distribution of rotational errors. However, their assumption
that the rotational error about each orthogonal axis has the same distribution is not generally true. Fitzpatrick and West [36], in 2001, derived an approximation of the TRE distribution that properly accounts for this anisotropy in the rotational error distributions using the Monte-Carlo simulation. In their derivation, the mean-squared value of TRE is also estimated for any arbitrary target location.

All the mentioned algorithms assume that FLE perturbing the fiducials has an isotropic and identical zero-mean Gaussian distribution. In 2007, Ma et al. [71] argued that, in real applications, FLE might be best modeled by identical and anisotropic Gaussian noise. Based on the spatial stiffness theory, they derived a solution which estimates the mean-squared value of TRE in the presence of this noise distribution. Seilhorst et al. [114], also in 2007, utilized the transformation of covariance method proposed in Hoff and Vincent [46], to estimate the TRE characteristics when FLE has inhomogeneous and anisotropic (heterogeneous) Gaussian distribution. Similarly, Wiles et al. [133], in 2008, modified the Fitzpatrick and West algorithm to derive another closed-form solution to estimate the mean-squared value of TRE in the presence of identical and anisotropic zero-mean Gaussian FLE. Their algorithm is based on a least-mean-squares solution to the fiducial registration problem. Their method can be extended by using a weighted least-mean-squares algorithm which is a more optimal solution for an identical and anisotropic FLE.

To the best of our knowledge, no closed-form solution has yet been developed that approximates the distribution of TRE, and calculates the mean-squared value of TRE at any desired point when FLE has a heterogenous (anisotropic and inhomogeneous) distribution. In this dissertation, we will propose the first general closed-form solution that estimates the distribution of TRE and its mean-squared value without using any
numerical simulations. We will also show that the proposed solution, in special cases
where FLE has an identical and isotropic or inhomogeneous and anisotropic zero-
mean Gaussian distribution, can be simplified to the ones previously presented in the
literature.

2.4 Summary

Pairwise point-based registration in the presence of isotropic and identical Gaussian
FLE is a well-understood problem in the literature and has several different closed-
form solutions. When FLE has anisotropic and inhomogeneous distribution, the
proposed closed-form solutions are suboptimal, and therefore, there is a need for a
new registration algorithm to optimally align two sets of corresponding points under
such noise conditions.

Pairwise surface-based registration in the presence of isotropic and identical zero-
mean Gaussian FLE is also intensively studied in the literature. Most of the presented
algorithms have focused on making the registration less susceptible to the initial
alignment error and noise. Little research has studied the impact of using a sequential
weighted least-mean-squares algorithm (such as UKF) in the registration process
when FLE has an arbitrary Gaussian distribution and initial alignment error between
data sets is relatively large.

In the groupwise point-based registration, due to the complexity of cost functions,
there is no closed-form solution has, as yet, been presented in the literature. Most
of the presented algorithms are based on an iterative algorithm which assumes that
FLE has an isotropic and identical zero-mean Gaussian distribution. Therefore, a
new iterative registration algorithm is proposed to simultaneously register a set of
corresponding points in the presence of FLE with an arbitrary Gaussian distribution.

Groupwise surface-based registration in the presence of isotropic and identical zero-mean Gaussian FLE has been also well studied in the literature for a decade. Most of the presented algorithms are based on an iterative pairwise surface-based registration method that assumes FLE perturbing data sets has an isotropic and identical distribution. Little work in the literature has investigated the effect of a sequential weighted least-mean-squares algorithm (such as UKF) in registering a set of surfaces when FLE has an arbitrary Gaussian distribution.

Finally, simulations studying the accuracy of the performed registration have been independently performed by several researchers, and analytical expressions for TRE have supported these results. However, relatively little research has focused on studying the behavior of the distribution of TRE and the mean-squared value of TRE when FLE has an arbitrary Gaussian distribution.
Chapter 3

Kalman Filtering and its Extensions

Before we explain the proposed registration algorithm based on the UKF, it would be useful to understand the concept of this technique. Therefore, the rest of this chapter will provide an overview on the optimal linear estimator, \textit{i.e.}, Kalman Filter, and its extensions.

In 1960, R. E. Kalman published his famous paper describing a recursive and incremental solution to the discrete data linear filtering problem [55]. Since then, the Kalman Filter has been the subject of extensive research and multiple applications, specifically in the area of navigation and target tracking [130]. The Kalman Filter is an optimum recursive algorithm that estimates a desired parameter (state vector), which is governed by a system called process model from observations computed from a linear system called an observation model.

The Kalman Filter incorporates all provided information about the noise in the system to estimate a state vector. Since it is a recursive algorithm, it can also adjust
itself if the provided information from the observation model changes. We will see, later on, how this property can be potentially useful in clinical applications of our proposed algorithm.

The Kalman Filter is optimum for linear models; however, registration is a non-linear process, and therefore, extensions of the Kalman Filter algorithm must be employed in solving the registration problems. In what follows, we first explain the Kalman Filter algorithm and then we discuss extensions, the Extended and Unscented Kalman Filter algorithms, respectively.

### 3.1 Kalman Filter

The Kalman Filter addresses the state vector estimation, $\mathbf{x}_i \in \mathbb{R}^n$, of a discrete-time control process model governed by a linear equation as

$$
\mathbf{x}_i = A\mathbf{x}_{i-1} + \mathbf{n}^x_{i-1},
$$

from the observation model, which is a linear function, as

$$
\mathbf{y}_i = C\mathbf{x}_i + \mathbf{n}^y_i,
$$

where $A$ and $C$ are defined by the system dynamics, $\mathbf{y}_i \in \mathbb{R}^m$ is the observation vector at time $i$, $\mathbf{n}^x_i$ and $\mathbf{n}^y_i$ represent the process and the observation noise at time $i$, respectively, and are independent Gaussian random variables with distributions $\mathcal{N}(0, \Sigma^x_i)$ and $\mathcal{N}(0, \Sigma^y_i)$. The Kalman Filter algorithm sequentially estimates the state vector $\mathbf{x}$ by minimizing its mean-squared error. This algorithm, by assuming to have initial values for the state vector $\mathbf{x}$, $\hat{\mathbf{x}}_0$, and its covariance matrix, $\mathbf{P}_{\hat{\mathbf{x}}_0}$, comprises two main steps.

In the first step, the filter estimates the state vector $\mathbf{x}$ and its covariance matrix
from the process model, Equation (3.1), as follows:

\[
\hat{x}_i^- = E[x_i] = A E[x_{i-1}] + E[n_{i-1}^x] = A \hat{x}_{i-1},
\]

(3.3)

\[
P_{\hat{x}_i^-} = E[(x_i - \hat{x}_i^-)(x_i - \hat{x}_i^-)'] = A E[(x_{i-1} - \hat{x}_{i-1})(x_{i-1} - \hat{x}_{i-1})']A^T + \Sigma_{i-1}^x,
\]

(3.4)

In the second step, the filter uses the observation vector and the observation model to update its estimated state vector and covariance matrix in the previous step as follows:

\[
\hat{x}_i = \hat{x}_i^- + K_i(y_i - \hat{y}_i^-),
\]

(3.5)

\[
P_{\hat{x}_i} = E[(x_i - \hat{x}_i)(x_i - \hat{x}_i)'] = (I - K_iC)P_{\hat{x}_i^-},
\]

(3.6)

where \(\hat{y}_i^- = C \hat{x}_i^-\), and \(K_i\) is called Kalman gain which is calculated as follows:

\[
K_i = \frac{P_{x_i y_i}}{P_{y_i}} = \frac{E[(x_i - \hat{x}_i^-)(y_i - \hat{y}_i^-)']}{E[(y_i - \hat{y}_i^-)(y_i - \hat{y}_i^-)']} = \frac{P_{x_i}^- C^T(CP_{x_i}^- C^T + \Sigma_y^-)^{-1}}{P_{y_i}}.
\]

(3.7)

This estimation is optimal if the process and the observation models are defined by linear equations, and if the process and observation noise are independent Gaussian random variables. However, in a general case, the process and the observation models can be governed by nonlinear equations as

\[
x_i = f(x_{i-1}, n_{i-1}^x),
\]

(3.8)

\[
y_i = h(x_i, n_i^y),
\]

(3.9)

where \(f\) and \(h\), in contrast to the previous case, are two nonlinear functions. In this situation, the Kalman Filter estimate of the state vector is not optimized since \(P_{x_i}^-\), \(P_{x_i y_i} = E[(x_i - \hat{x}_i^-)(y_i - \hat{y}_i^-)']\), and \(P_{y_i} = E[(y_i - \hat{y}_i^-)(y_i - \hat{y}_i^-)']\) cannot be computed in closed-form as before. Two well-known solutions exist that deal with nonlinearities in the process or observation model.
3.2 Extended Kalman Filter

In the first solution, a nonlinear function in the process or observation model can be linearized around a good initial guess, using the first-order Taylor series expansion, as

\[
\begin{align*}
\mathbf{x}_i &\approx \tilde{\mathbf{x}}_i + A(\mathbf{x}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + B\mathbf{n}_{i-1}^x, \\
\mathbf{y}_i &\approx \tilde{\mathbf{y}}_i + C(\mathbf{x}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + D\mathbf{n}_{i}^y,
\end{align*}
\]

where \( \tilde{\mathbf{x}}_i = f(\tilde{\mathbf{x}}_{i-1}, 0) \), \( \tilde{\mathbf{y}}_i = h(\tilde{\mathbf{x}}_i, 0) \), \( A = \frac{\partial f(\tilde{\mathbf{x}}_{i-1}, 0)}{\partial \mathbf{x}} \), \( C = \frac{\partial h(\tilde{\mathbf{x}}_i, 0)}{\partial \mathbf{x}} \), \( B = \frac{\partial f(\tilde{\mathbf{x}}_{i-1}, 0)}{\partial \mathbf{n}_i^x} \), and \( D = \frac{\partial h(\tilde{\mathbf{x}}_i, 0)}{\partial \mathbf{n}_i^y} \).

Then, the Kalman Filter algorithm can be used to estimate the state vector as described above. This method is called EKF [130]. However, to use EKF, one needs to have the first derivative of the nonlinear functions (Jacobian matrices) to linearize the nonlinearities. Finding the Jacobian matrix – if it exists – is usually cumbersome and adds to the complexity of the algorithm.

A second approach, called UKF [54], utilizes an Unscented Transform (UT) [53] and the true nonlinear models to approximate the distribution of the state vector (mean and covariance matrix of the state vector) rather than linearizing the nonlinear process or observation model. In the next section, first the Unscented Transform is described and afterwards the UKF algorithm based on the Unscented Transform is presented.

3.3 Unscented Transform

The Unscented Transform (UT) is a method for computing the characteristics of a random variable which undergoes a nonlinear transformation. Consider propagating
an \( n_x \) dimensional random variable \( \mathbf{x}_i \) through an arbitrary nonlinear function \( h \) to generate an \( n_y \) dimensional random variable \( \mathbf{y}_i \) as

\[
\mathbf{y}_i = h(\mathbf{x}_i). \tag{3.12}
\]

Let us assume that \( \mathbf{x}_i \) has the mean \( \hat{\mathbf{x}}_i \) and covariance matrix \( \mathbf{P}_{x_i} \). To compute \( \hat{\mathbf{y}}_i = \mathbb{E}[\mathbf{y}_i] \) and \( \mathbf{P}_{y_i} = \mathbb{E}[(\mathbf{y}_i - \hat{\mathbf{y}}_i)(\mathbf{y}_i - \hat{\mathbf{y}}_i)^T] \), one should choose a set of \( 2n_x + 1 \) weighted sigma points \( s = \{w^m_k, w^c_k, \tilde{x}_{k,i}\} \) (where \( \tilde{x}_{k,i} \) is a sigma point and \( w^m_k \) and \( w^c_k \) are the weights associated with it) so that they completely capture the mean and covariance matrix of the random variable \( \mathbf{x}_i \) as follows [78]:

\[
\tilde{x}_{0,i} = \hat{\mathbf{x}}_i \quad k = 0, \tag{3.13}
\]

\[
\tilde{x}_{k,i} = \hat{\mathbf{x}}_i + \sqrt{(n_x + \lambda)\mathbf{P}_{x_i}}_k \quad k = 1, \ldots, n_x, \tag{3.14}
\]

\[
\tilde{x}_{n_x + 1,i} = \hat{\mathbf{x}}_i - \sqrt{(n_x + \lambda)\mathbf{P}_{x_i}}_k \quad k = n_x + 1, \ldots, 2n_x, \tag{3.15}
\]

\[
w^m_0 = \frac{\lambda}{\lambda + n_x} \quad k = 0, \tag{3.16}
\]

\[
w^c_0 = \frac{\lambda}{\lambda + n_x} + (1 - \alpha^2 + \beta) \quad k = 0, \tag{3.17}
\]

\[
w^m_k = w^c_k = \frac{1}{2(\lambda + n_x)} \quad k = 1, \ldots, 2n_x, \tag{3.18}
\]

where \( \lambda = \alpha^2(n_x + \kappa) - n_x \), and \( \sqrt{(n_x + \lambda)\mathbf{P}_{x_i}}_k \) is the \( k \)th row or column of matrix square root of \( (n_x + \lambda)\mathbf{P}_{x_i} \). \( \kappa \geq 0 \) is chosen to guarantee positive semi-definiteness of \( \mathbf{P}_{y_i} \). \( 0 \leq \alpha \leq 1 \) controls the size of the sigma points’ distribution and should ideally be a small value. \( \beta \geq 0 \) is a nonnegative weighting term which can be used to incorporate knowledge of the higher order moments of the distribution (\( \kappa, \alpha \) and \( \beta \) parameters might be chosen as 500, 0.25, and 3, respectively, as suggested in [78]). The calculated \( 2n_x + 1 \) sigma points are then propagated through the nonlinear transformation as

\[
\tilde{\mathbf{y}}_{k,i} = h(\tilde{\mathbf{x}}_{k,i}) \quad k = 0, \ldots, 2n_x. \tag{3.19}
\]
Finally, \( \hat{y}_i \) and \( P_{y_i} \) are computed as follows:

\[
\hat{y}_i = \sum_{k=0}^{2n_x} w^y_m^k \tilde{y}_{k,i}, \tag{3.20}
\]

\[
P_{y_i} = \sum_{k=0}^{2n_x} w^c_k (\tilde{y}_{k,i} - \hat{y}_i)(\tilde{y}_{k,i} - \hat{y}_i)^T. \tag{3.21}
\]

### 3.4 Unscented Kalman Filter

UKF is a straightforward application of the Unscented Transform. The UKF algorithm that computes the mean and covariance matrix of the state vector \( x_i \) governed by a nonlinear process model, Equation (3.8), and a nonlinear observation model, Equation (3.9), is given as follows (similar notations to [128] have been used):

1) Define the state random variable \( x^a_i \) as the concatenation of the original state and noise variables as: \( x^a_i = [x^T_i, n^x_i, n^y_i]^T \equiv [x^x_i, x^y_i, n^x_i, n^y_i]^T \) and initialize with

\[
\hat{x}_0 = E[x_0], \tag{3.22}
\]

\[
\hat{x}^a_0 = E[x^T_0, n^x_0, n^y_0]^T = [\hat{x}^x_0, 0, 0]^T, \tag{3.23}
\]

\[
P_{\hat{x}_0} = E[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T], \tag{3.24}
\]

\[
P_{\hat{x}^a_0} = E[(x^a_0 - \hat{x}^a_0)(x^a_0 - \hat{x}^a_0)^T] = \text{diag}(P_{\hat{x}_0}, \Sigma^x_0, \Sigma^y_0). \tag{3.25}
\]

2) Calculate the sigma points as

\[
\tilde{x}_{0,i} = \hat{x}^a_i, \tag{3.26}
\]

\[
\tilde{x}_{k,i} = \hat{x}^a_i + (\sqrt{(n_a + \lambda)P_{\hat{x}^a_i}})_k \quad k = 1, \ldots, n_a, \tag{3.27}
\]

\[
\tilde{x}_{k,i} = \hat{x}^a_i - (\sqrt{(n_a + \lambda)P_{\hat{x}^a_i}})_k \quad k = n_a + 1, \ldots, 2n_a. \tag{3.28}
\]
3) Propagate the sigma points through the process and observation models to estimate the means and covariance matrices as

\[ \tilde{x}_{k,i} = f(\tilde{x}_{k,i}, \tilde{x}_{n,k}) , \]  
\[ \tilde{x}_i = \sum_{k=0}^{2n_a} w_k^m \tilde{x}_{k,i} , \]  
\[ P_{\tilde{x}_i} = \sum_{k=0}^{2n_a} w_k^c [\tilde{x}_{k,i} - \tilde{x}_i] [\tilde{x}_{k,i} - \tilde{x}_i]^T , \]  
\[ \tilde{y}_{k,i} = h(\tilde{x}_{k,i}, \tilde{x}_{n,y_k}) , \]  
\[ \tilde{y}_i = \sum_{k=0}^{2n_a} w_k^m \tilde{y}_{k,i} . \]

4) Update the measurements as follows:

\[ P_{y_i} = \sum_{k=0}^{2n_a} w_k^c [\tilde{y}_{k,i} - \tilde{y}_i] [\tilde{y}_{k,i} - \tilde{y}_i]^T , \]  
\[ P_{x,y_i} = \sum_{k=0}^{2n_a} w_k^c [\tilde{x}_{x_{k,i}} - \tilde{x}_i] [\tilde{y}_{k,i} - \tilde{y}_i]^T , \]  
\[ K_i = P_{x,y_i} P_{y_i}^{-1} , \]  
\[ \tilde{x}_i = \tilde{x}_i + K_i (y_i - \tilde{y}_i) , \]  
\[ P_{\tilde{x}_i} = P_{\tilde{x}_i} - K_i P_{y_i} K_i^T . \]

3.5 Summary

The Kalman Filter algorithm is an optimum recursive algorithm that estimates the desired parameters (state vector \( \mathbf{x} \)) governed by a linear process model, from a linear observation model. If either the process or observation model is nonlinear, then the Kalman Filter algorithm is not optimum, and its extensions should be used to estimate the state vector. To deal with these nonlinearities, EKF and UKF algorithms have
been proposed in the literature.

In the following chapters, we will discuss how the UKF algorithm can be used to register two or multiple data sets, and how its output can be utilized to estimate the accuracy of the performed registration.
Chapter 4

UKF Pairwise Registration

In this chapter, we use the theory discussed in Chapter 3 to present two novel registration algorithms based on the Unscented Kalman Filter (UKF). First, we propose a UKF point-based registration algorithm to align two sets of corresponding points in the presence of FLE with an arbitrary zero-mean Gaussian distribution. Then, we extend the proposed method to a surface-based registration algorithm where the corresponding points between surfaces are not known but the surfaces are roughly aligned. The main results of this chapter are

1) The performance of the proposed UKF point-based registration algorithm in the presence of identical isotropic zero-mean Gaussian FLE equates to those in Umeyama [126] and Horn [47, 48].

2) The proposed UKF point-based registration algorithm outperforms its peer algorithm, the EKF registration algorithm [98], in registering two data sets in the presence of identical isotropic zero-mean Gaussian noise.

3) The proposed UKF point-based registration algorithm more robustly registers two data sets than the Umeyama [126] and Arun [2] algorithms, when FLE has an
anisotropic inhomogeneous distribution.

4) The proposed UKF surface-based registration algorithm more robustly and accurately registers two surfaces than the well-known standard ICP registration algorithm.

4.1 UKF Pairwise Point-based Registration

Here, the goal is to determine a $3 \times 3$ rotation matrix $R$ and a $3 \times 1$ translation vector $t$ that register two sets of corresponding points $U$ and $Y$, from the following nonlinear function

$$y_i = Ru_i + t + n_i^y, \quad i = 1, ..., N,$$

such that the following cost function (mean-squared distance error between the two data sets, $U$ and $Y$) is recursively minimized [117]

$$C(R, t) = \frac{1}{N} \sum_{i=1}^{N} (y_i - Ru_i - t)^T (\Sigma_i^y)^{-1} (y_i - Ru_i - t).$$

To estimate the rotation matrix $R$ and translation vector $t$ in (4.1), $R$ is represented by the Euler angles [110] as follows:

$$R(\theta_x, \theta_y, \theta_z) = R(\theta_z) \times R(\theta_y) \times R(\theta_x),$$

where $\theta_x$, $\theta_y$ and $\theta_z$ are the rotation angles around orthogonal coordinate frame axes ($x$, $y$ and $z$). In this thesis, the Euler angles are utilized to represent the rotation matrix $R$, and we have accounted for this representation’s singularity points in our simulations.

The translation vector $t$ is defined as $[t_x, t_y, t_z]^T$, where $t_x$, $t_y$ and $t_z$ are the translational parameters along $x$, $y$ and $z$ axes, respectively. Let us define the state
vector \( \mathbf{x} \) as a \( 6 \times 1 \) vector, including the registration transformation parameters, as
\[
\mathbf{x} = [t_x, t_y, t_z, \theta_x, \theta_y, \theta_z]^T = [\mathbf{x}_t^T, \mathbf{x}_\theta^T]^T, \tag{4.4}
\]
where \( \mathbf{x}_t^T \) and \( \mathbf{x}_\theta^T \) are defined to be \([t_x, t_y, t_z]\) and \([\theta_x, \theta_y, \theta_z]\), respectively. In addition, let us define the process model as follows:
\[
\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{n}_i^x, \tag{4.5}
\]
where \( \mathbf{x} \) is the state vector in (4.4) with the initial value \( \hat{\mathbf{x}}_0 \) and covariance matrix \( \mathbf{P}_{\mathbf{x}_0} \). \( \mathbf{n}_i^x \) is a zero-mean Gaussian random vector with distribution \( \mathbf{\Sigma}_i^x, \mathcal{N}(0, \mathbf{\Sigma}_i^x) \). Here, it is assumed that the state vector \( \mathbf{x} \) possesses a Gaussian distribution.

Now, we wish to estimate the state vector \( \mathbf{x} \) from (4.1) using the Kalman Filter algorithm. However, the problem with (4.1) is that both the measurement point \( \mathbf{y}_i \) and the input point \( \mathbf{u}_i \) are perturbed by Gaussian noise. To fit this problem to the Kalman Filter framework where the input points are deterministically known and the observation points are noisy, we reformulate (4.1) as follows.

By substituting (2.2) into (4.1), (4.1) can be rewritten as
\[
\mathbf{y}_i = \mathbf{R}_{(\mathbf{x}_0)} \hat{\mathbf{u}}_i + \mathbf{x}_t + \mathbf{R}_{(\mathbf{x}_0)} \mathbf{n}_i^u + \mathbf{n}_i^y, \tag{4.6}
\]
where \( \mathbf{x}_t \) and \( \mathbf{x}_\theta \) are defined in (4.4). If \( \mathbf{n}_i^u \) is assumed to be isotropic zero-mean Gaussian random noise with distribution \( \mathcal{N}(0, \mathbf{\Sigma}_i^u) \), \( \mathbf{R}_{(\mathbf{x}_0)} \mathbf{n}_i^u \) is also an isotropic zero-mean Gaussian random vector with covariance matrix \( \mathbf{\Sigma}_i^u \). Furthermore, because \( \mathbf{R}_{(\mathbf{x}_0)} \mathbf{n}_i^u \) and \( \mathbf{n}_i^y \) are zero-mean independent Gaussian noise vectors, their summation is also a zero-mean Gaussian noise vector. Therefore, (4.6) can be rewritten as
\[
\mathbf{y}_i = \mathbf{R}_{(\mathbf{x}_0)} \hat{\mathbf{u}}_i + \mathbf{x}_t + \mathbf{n}_i^y = \mathbf{\hat{y}}_i + \mathbf{n}_i^y, \tag{4.7}
\]
where \( \mathbf{\hat{y}}_i = \mathbf{R}_{(\mathbf{x}_0)} \hat{\mathbf{u}}_i + \mathbf{x}_t \), and \( \mathbf{n}_i^y = \mathbf{R}_{(\mathbf{x}_0)} \mathbf{n}_i^u + \mathbf{n}_i^y \) is a zero-mean Gaussian random
vector with the modified covariance matrix

$$\Sigma^\prime_y = \Sigma^u_y + \Sigma^y_i.$$  (4.8)

If \(n^u_i\), and \(n^y_i\) are two independent isotropic zero-mean Gaussian noise with distributions \(N(0, \sigma^2_{i,u} I)\), and \(N(0, \sigma^2_{i,y} I)\), respectively, then (4.8) can be simplified as

$$\Sigma^\prime_y = (\sigma^2_{i,u} + \sigma^2_{i,y})I_{3\times3} = \sigma^2_i I_{3\times3}. \quad (4.9)$$

Equations (4.7) and (4.9) simply mean that if data sets \(Y\) and \(U\) are corrupted by isotropic zero-mean Gaussian random noise, then, without loss of generality, one can assume that the additive Gaussian noise in data set \(U\) is zero by adding its variance to the variance of the noise in data set \(Y\) [61].

Equation (4.7) helps us reformulate the original registration problem – Equation (4.1) – as an observation model where the measurements (the points in the fixed data set) have all the noise, and the points in the moving data set are deterministically known. Therefore, one could use the Kalman Filter algorithm to estimate the registration transformation parameters. By using (4.7), the state vector \(x\) is estimated from the following nonlinear function (also called the observation model)

$$y_{1:i} = \tilde{y}_{1:i} + n^\prime_{1:i}, = R(x_i) \tilde{u}_{1:i} + [x_1, x_2, ..., x_i] + n^\prime_{1:i}, \quad i = 1, ..., N,$$  (4.10)

where \(y_{1:i} = [y_1^T, ..., y_i^T]^T_{3i\times1}\), \(\tilde{y}_{1:i} = [\tilde{y}_1^T, ..., \tilde{y}_i^T]^T_{3i\times1}\), and \(n^\prime_{1:i} = [n^\prime_1^T, ..., n^\prime_i^T]^T_{3i\times1}\). \(n^\prime_{1:i}\) is a zero-mean Gaussian noise with covariance matrix \(\Sigma^\prime_{1:i}\), which models the fiducial localization error (FLE) incurred by collecting points from rigid objects in both fixed and moving data sets.

Assuming that every point in the moving and fixed data sets is perturbed by isotropic zero-mean Gaussian noise with covariance matrices \(\sigma^2_{i,u} I_{3\times3}\) and \(\sigma^2_{i,y} I_{3\times3}\), then \(n^\prime_{1:i}\) is a zero-mean Gaussian random vector with covariance matrix \(\Sigma^\prime_{1:i} = \)
CHAPTER 4. UKF PAIRWISE REGISTRATION

$\text{diag}(\Sigma_1^y, \ldots, \Sigma_i^y)$, where $\Sigma_i^y$ is defined in (4.9).

It is important to mention that, in the observation model, Equation (4.10), $i$ observations, $y_1:i$, have been used to estimate the state vector $x$ rather than using the most recent one, $y_i$. Although if one uses the latter observation the algorithm still converges to the true solution, later on we will show that the computed variance of the state vector $x$ is unreliable and does not fit those generated by numerical simulations. The more observations one takes into account, the more reliable the reported variances are and the more closely they match those generated by numerical simulations. Since the estimated variances carry information that can be used in further applications, we have employed $i$ observation, $y_{1:i}$, to estimate the state vector $x$.

Since (4.10) is nonlinear in terms of the state vector $x$, the EKF algorithm may be utilized to estimate $x$ [98]. Consequently, one needs to linearize the observation model around an initial vector for $x$ by using the first-order Taylor series expansion, and to use the standard Kalman Filter algorithm [55] to find $x$. However, since the model is highly nonlinear (in terms of the rotational parameters $\theta_x$, $\theta_y$, and $\theta_z$), the local linearity assumption easily shatters when the initial state vector is insufficiently close to its optimal value. In this situation, more Taylor series terms should be considered in the linearized model.

To overcome these limitations, we utilize the UKF algorithm as a derivative-free alternative to the EKF algorithm. The UKF algorithm uses true nonlinear models to approximate the distribution of an observation vector with the accuracy of at least a second-order Taylor series expansion. The EKF algorithm, in contrast, only achieves the accuracy of the first-order Taylor series expansion, with the same order of complexity as that of the UKF algorithm [128]. In the following sections, we describe
the proposed UKF point-based registration algorithms when FLE has isotropic and anisotropic zero-mean Gaussian FLE, respectively.

4.1.1 Isotropic FLE Distribution

To estimate the state vector $\mathbf{x}$ – governed by a linear process model, Equation (4.5) – from a nonlinear observation model, Equation (4.10), the following algorithm iterates until it converges to a final solution (see Figure 4.1).

![Flowchart of the UKF registration algorithm in the presence of isotropic zero-mean Gaussian noise when correspondences between two data sets are known.](image)

1) Initialize the state vector $\mathbf{x}$ and its covariance matrix with zero and $\mathbf{I}_{6 \times 6}$, respectively. Let the covariance of the observation noise be the covariance of the additive Gaussian noise in the two data sets, $\mathbf{\Sigma}_i^{y}$. Also, let the covariance of the process model, $\mathbf{\Sigma}_i^{x}$ which is heuristically determined, be
\[ \Sigma_i^x = \text{diag}(\sigma_i^2(x), \sigma_i^2(y), \sigma_i^2(z)), \quad \frac{c^2}{\sqrt{\lambda^2(x) \sigma_i^2(z)}} + \frac{c^2}{\sqrt{\lambda^2(y) \sigma_i^2(z)}} + \frac{c^2}{\sqrt{\lambda^2(z) \sigma_i^2(x)}} \]  
(4.11)

where \( \lambda^2(x), \lambda^2(y) \) and \( \lambda^2(z) \) are variances of the fixed data set along \( x, y \) and \( z \) axes, respectively; \( \sigma_i^2(x), \sigma_i^2(y) \) and \( \sigma_i^2(z) \) are the diagonal elements of the covariance matrix \( \Sigma_i^y \) (they are the variances of the zero-mean Gaussian noise added to the fixed data set along \( x, y \) and \( z \) axes, respectively); and \( c \) is a constant which, in our simulations, is set to be \( \sqrt{180} \). It is assumed that the variance of noise along each axis has a direct effect on the variance of the translational parameter along that axis. Furthermore, the variance of the rotational parameter along an axis is approximately inversely proportional to the ratio of a) the variances of the data set along the other two orthogonal axes, and b) the variances of noise along those axes.

The intuition is founded on the fact that if the variance of noise along an axis is zero, then the orientation of the other two orthogonal axes can be precisely determined. Equation (4.11) has been proposed to satisfy the above assumptions. In this equation, if the variance of the noise along an axis is zero, the covariance matrix element for the corresponding translational parameter is zero. Furthermore, the elements that correspond to the rotational parameters orthogonal to that axis would be zero as well. Constant \( c \) is a mere normalization factor, and is determined by numerical simulations so that the computed variances by the UKF registration algorithm match those derived by numerical simulations.

2) Predict the state vector \( \mathbf{x} \) and its covariance matrix from the process model,
Equation (4.5), as
\[
\hat{x}_i^- = \hat{x}_{i-1},
\]
\[
P_{\hat{x}_i^-} = P_{\hat{x}_{i-1}} + \Sigma_{i-1}^x,
\]
where \(\hat{x}_i^-\) and \(P_{\hat{x}_i^-}\) are the predicted state vector and the predicted covariance matrix, respectively.

3) Append the \(i\)th point from the moving data set \(U\) to the set of previously selected points from that data set and estimate the corresponding points' positions, \(\hat{y}_{1:i}^- = [\hat{y}_{1}^-, \ldots, \hat{y}_i^-]\), in the fixed data set \(Y\) by using the predicted state vector from Step 2 as follows:
\[
\hat{y}_i^- = R(\hat{x}_i^-)u_i + \hat{x}_i^-.
\]

4) Compute the error in the estimated corresponding points in data set \(Y\), \((y_{1:i} - \hat{y}_{1:i}^-)\), to update the state vector and its covariance matrix as follows:
\[
\hat{x}_i = \hat{x}_i^- + K_i(y_{1:i} - \hat{y}_{1:i}^-),
\]
\[
P_{\hat{x}_i} = P_{\hat{x}_i^-} - K_iP_{y_{1:i}}K_i^T,
\]
where \(K_i = P_{x,y_{1:i}}P_{y_{1:i}}^{-1}\) is the Kalman gain and \(P_{x,y_{1:i}}\) and \(P_{y_{1:i}}\) are defined to be \(E[(x_i - \hat{x}_i^-)(y_{1:i} - \hat{y}_{1:i}^-)^T]\) and \(E[(y_{1:i} - \hat{y}_{1:i}^-)(y_{1:i} - \hat{y}_{1:i}^-)^T]\), respectively. Since the observation model, Equation (4.10), is nonlinear, the method outlined in Section 3.4 is used to compute \(P_{x,y_{1:i}}\) and \(P_{y_{1:i}}\), respectively. This algorithm iterates through all the points in the moving data set \(U\).

4.1.2 Anisotropic FLE Distribution

In this section, we extend the proposed UKF point-based registration algorithm to register two data sets in the presence of anisotropic zero-mean Gaussian FLE. Assume
that the additive noise for the points with index $i$ in the fixed data set $Y$ and, in the moving data set $U$, is independent anisotropic zero-mean Gaussian noise with covariance matrices $\Sigma_y^i$ and $\Sigma_u^i$, respectively. Consequently, $n_y^i = R(\theta(x))n_u^i + n_y^i$ is a zero-mean Gaussian random vector with covariance matrix

$$
\Sigma_y^i = E[n_y^i n_y^i]^T = E[R(\theta(x))n_u^i + n_y^i][R(\theta(x))n_u^i + n_y^i]^T = E[R(\theta(x))n_u^i n_u^i]^T + E[n_y^i n_y^i]^T = R(\theta(x))\Sigma_u^i R^T(\theta(x)) + \Sigma_y^i. \quad(4.17)
$$

It can be seen from (4.17) that $n_y^i$ is a correlated anisotropic noise whose covariance matrix depends on the rotation matrix $R(\theta(x))$.

Again, one can assume that the additive Gaussian noise in the moving data set $U$ is zero by adding its covariance matrix to the covariance matrix of the noise in the fixed data set $Y$. Then, the registration transformation parameters can be estimated as before by using (4.10) as the observation model. However, the modified covariance matrix of the noise in the fixed data set, $\Sigma_y^i$, is not known in advance. In order to use the UKF registration algorithm, one needs to somehow estimate the covariance matrix $\Sigma_y^i$. We utilize the computed rotational parameters in each iteration of the UKF registration algorithm to estimate $\Sigma_y^i$, as follows (Figure 4.2):

1) Initialize the state vector $x$ and its covariance matrix with zero and the identity matrix, respectively; Initialize the covariance matrix of the state vector using (4.11); Use (4.17) to find the variance of the noise added to the fixed data set along the $x$, $y$, and $z$ axes. Given that the rotation matrix is not known in advance, it is replaced with the identity matrix in order to calculate the variance of the noise along each axis.

2) Predict the state vector $x$ and its covariance matrix from the process model,
Figure 4.2: Flowchart of the UKF registration algorithm in the presence of anisotropic zero-mean Gaussian noise when correspondences between two data sets are known.

Equation (5.2), as

\[
\hat{x}_i^- = \hat{x}_{i-1}, \tag{4.18}
\]

\[
P_{x_i}^- = P_{\hat{x}_{i-1}} + \Sigma_{x{i-1}}^x, \tag{4.19}
\]

3) Append the \( i \)th point from the moving data set \( U \) to the set of previously selected points from that data set and estimate the corresponding points’ positions, \( \hat{y}_i^- = [\hat{y}_1^-, ..., \hat{y}_i^-] \), in the fixed data set \( Y \) by using the predicted state vector from Step 2 as follows:

\[
\hat{y}_i^- = R_{\hat{y}^-} u_i + \check{x}_t^- \tag{4.20}
\]

4) Update the modified covariance matrix \( \Sigma^y_x \) using the predicted state vector in
CHAPTER 4. UKF PAIRWISE REGISTRATION

Step 2, as

\[ \hat{\Sigma}_i^y = R_{\hat{x}_i}(\hat{x}_i) \Sigma_i^y R_{\hat{x}_i}^T + \Sigma_i^y. \]  

(4.21)

5) Use \( \hat{\Sigma}_i^y \) to calculate \( P_{y,1,i} \) and \( P_{x,y,1,i} \), respectively, as explained in Section 3.4. Then, compute the error in the estimated corresponding points in the data set \( Y \), \( (y_{1,i} - \hat{y}_{1,i}) \), to update the state vector and its covariance matrix, as follows:

\[ \hat{x}_i = \hat{x}_i^- + K_i (y_{1,i} - \hat{y}_{1,i}), \]  

(4.22)

\[ P_{\hat{x}_i} = P_{\hat{x}_i}^- - K_i P_{y,1,i} K_i^T. \]  

(4.23)

The procedure iterates through all the points in the moving data set \( U \) to converge on a solution.

In the following section, the performance of the proposed registration algorithms in Sections 4.1.1 and 4.1.2 is verified on randomly generated data sets in the presence of isotropic and anisotropic zero-mean Gaussian FLE. To perform the simulations, MATLAB™ 7.0.4 software from MathWorks, running on a 3.00GHz Pentium 4CPU desktop with 512MB RAM, is used.

4.1.3 Numerical Simulations

In this section, multiple numerical simulations are performed to examine and validate the success of the proposed algorithms. Simulations (a) and (b) compare the performance of the proposed algorithm with the Horn [48] and Umeyama [126] algorithms in registering two data sets when FLE has an identical and inhomogeneous isotropic zero-mean Gaussian distribution, respectively. Simulation (c) validates the proposed algorithm in registering two data sets perturbed by identical and inhomogeneous anisotropic zero-mean Gaussian FLE. The behavior of the proposed algorithm
CHAPTER 4. UKF PAIRWISE REGISTRATION

with respect to the order of the points used in the UKF algorithm is examined in Simulation 4. Simulation 5 illustrates the sensitivity of the proposed algorithm with respect to the variance of FLE in the registering data sets. The sensitivity of the proposed algorithm with respect to the number of points used in UKF is investigated in Simulation 5. Simulation 6 analyzes the sensitivity of the proposed algorithm to the number of observations in UKF. Finally, the proposed UKF registration algorithm is compared with the EKF registration algorithm [98] in Simulation 7.

a) Performance of the proposed algorithm in the presence of identical anisotropic zero-mean Gaussian noise:

In this simulation, we produce the noiseless three-dimensional moving data set $\hat{\mathbf{U}}$ by drawing $N = 30^1$ points uniformly in the interval $[-250 \text{mm}, 250 \text{mm}]$. To create the noiseless fixed data set $\hat{\mathbf{Y}}$, a random transformation is applied to $\hat{\mathbf{U}}$. This transformation is generated by uniformly drawing the rotational and translational parameters in the intervals $[-90^\circ, 90^\circ]$ and $[-90 \text{mm}, 90 \text{mm}]$, respectively.

In this experiment, an isotropic zero-mean Gaussian random noise vector $\mathbf{n}_i^\theta$, with covariance of $\Sigma_i^\theta = (1 \text{mm}^2)\mathbf{I}_{3 \times 3}$, is added to each point in the data set $\hat{\mathbf{Y}}$ to generate the fixed data set $\mathbf{Y}$. Next, the UKF registration algorithm with random ordering of matching points in the data sets is utilized to register the moving data set $\hat{\mathbf{U}}$ to $\mathbf{Y}$. This procedure is repeated 5,000 times with different transformations and different data sets that are generated randomly.

To examine our registration algorithm, we also employ the methods proposed by Umeyama [126] and Horn et al. [48] to estimate the rotational and translational

---

^1In most of the performed numerical simulations, it is assumed that $N = 10$ or 30; Future work will explore the performance of the proposed technique when $N$ is less than 10.
parameters.

The simulation results show that all three algorithms accurately estimate the transformation parameters 97.5 percent of the time with the error of less than 1 mm and 1°, respectively. In the other 2.5 percent of the time, the algorithms need more sample points ($N > 30$) to attain the same accuracy.

Figure 4.3 depicts the error histogram of the estimated transformation parameters using the UKF and Umeyama registration algorithms, respectively. The mean and standard deviation (std) of error of the estimated transformation parameters are illustrated in Table 4.1. As can be seen from Table 4.1, the Umeyama, Horn and UKF registration algorithms obtain almost the same results\(^2\) by minimizing the same cost function $C$, Equation (4.2). However, the UKF registration algorithm iteratively minimizes the cost function $C$ by sequentially processing the points in the moving data set, whereas the registration algorithms proposed by Umeyama and Horn use all the points simultaneously to minimize $C$.

The advantage of the UKF registration algorithm over Umeyama and Horn algorithms is that the former also computes the variance of the estimated transformation parameters. Figure 4.4 depicts the transformation parameters and their standard deviations using the UKF registration algorithm in one of the simulation runs. In this run, the nominal translation and rotation parameters along $x$, $y$ and $z$ axes are $[50.96 \text{ mm}, -78.67 \text{ mm}, 70.71 \text{ mm}]$ and $[65.03^\circ, 5.98^\circ, -33.17^\circ]$, respectively. As shown, the algorithm converges to $[51 \text{ mm}, -78.7 \text{ mm}, 71 \text{ mm}]$ and $[64.9^\circ, 5.9^\circ, -33.2^\circ]$ for translational and rotational parameters after processing 30

\(^2\)In this thesis, we have used t-test and Levene’s test for equality of variances to verify the statistical significance of the mean and variance differences, respectively.
Figure 4.3: Error distribution of the estimated transformation parameters using UKF and Umeyama registration algorithms over 5,000 trials in the presence of isotropic Gaussian noise.
points. These points are ordered randomly and are fed to the UKF registration algorithm sequentially. Later on, it will be shown that the order of points is of no importance in the convergence of the proposed algorithm.

Another interesting result is that the variance of the estimated parameters can be directly and accurately computed by using the UKF registration algorithm. The computed variances represent a confidence interval for the estimated transformation parameters. Additional information that one can detect from Figure 4.4 is that the registration procedure can be stopped when the estimated transformation parameters reach the desired confidence interval; otherwise, more points should be given to the algorithm to reduce the variances.

The same results are derived by increasing the variance of $n_i^\theta$ from 1 $mm^2$ to 10 $mm^2$ with this difference: all the algorithms accurately estimate the transformation parameters, with errors of less than 1 $mm$ and 1 degree, only 77.5% percent of the time. In other words, if the variance of the noise in the datasets is increased, the number of registration points should also be increased to maintain a similar performance.

b) Performance of the proposed algorithm in the presence of inhomogeneous isotropic zero-mean Gaussian noise:

Here, we assign different isotropic zero-mean Gaussian distribution to each point in $\tilde{Y}$. The variance of these Gaussian distributions is randomly generated from a uniform distribution between 0 and 20 $mm^2$. The data sets and random transformations are generated in the same way as in the previous simulation. Table 4.2 shows the mean and standard deviation (std) of error of the estimated transformation parameters.
Figure 4.4: The estimated transformation parameters and their standard deviations using the UKF registration algorithm for registration of two randomly generated data sets in the presence of isotropic zero-mean Gaussian noise with covariance matrix $I_{3 \times 3}$. $N$, number of points in the data sets, is 30.
Table 4.1: Mean and standard deviation of error of the estimated translational (mm) and rotational (degree) parameters using the UKF, Umeyama and Horn registration algorithms in the presence of the isotropic zero-mean Gaussian noise with covariance matrix $I_{3\times3}$. The difference between the UKF technique and the other two methods fails to show statistical significance for any of the estimated parameters ($p > 0.05$).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>-0.00004</td>
<td>0.183</td>
<td>0.00006</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.0033</td>
<td>0.181</td>
<td>0.0033</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.0011</td>
<td>0.18</td>
<td>-0.0006</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>0.0021</td>
<td>0.154</td>
<td>0.002</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>0.0007</td>
<td>0.05</td>
<td>0.0006</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>-0.0015</td>
<td>0.151</td>
<td>-0.0005</td>
</tr>
</tbody>
</table>

using the UKF and Umeyama registration algorithms. As can be seen from Table 4.2, the UKF registration algorithm is able to more accurately register the two data sets in this situation. Also, we should add that, in this experiment, for 93 percent of the time, the UKF registration algorithm estimates the transformation parameters with an error of less than 1 mm and 1 degree; however, the Umeyama registration algorithm can only estimate the transformation parameters with that accuracy for 71 percent of the time.

c) Performance of the proposed algorithm in the presence of anisotropic zero-mean Gaussian noise:

In this experiment, the moving data set $\ddot{U}$ is generated as before; however, the additive noise in both data sets, $n^u_i$ and $n^y_i$, is assumed to be anisotropic zero-mean Gaussian
Table 4.2: Mean and standard deviation of error of the estimated translational (mm) and rotational (degree) parameters using the UKF and Umeyama registration algorithms. The difference between the mean errors for the two methods fails to show statistical significance for any of the estimated parameters ($p > 0.05$); however, there is statistically significant difference between the calculated standard deviations ($p < 0.05$).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>UKF Alg.</th>
<th>Umeyama Alg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>-0.005</td>
<td>0.32</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.002</td>
<td>0.33</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.005</td>
<td>0.33</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>0.00005</td>
<td>0.22</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>0.0007</td>
<td>0.10</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>0.002</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Noise with covariance matrices

\[
\Sigma^u_i = \begin{pmatrix}
2 & 0 & 0 \\
0 & 3 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}, \quad \Sigma^y_i = \begin{pmatrix}
30 & 0 & 0 \\
0 & 3 & 0 \\
0 & 0 & 1 \\
\end{pmatrix},
\]

for all the points in the data sets, respectively. To generate the data set $\tilde{Y}$, a random transformation is applied on the data set $\tilde{U}$. The transformation is produced by uniformly drawing the rotational and translational parameters in the intervals $[-90^\circ, 90^\circ]$ and $[-90 \text{ mm}, 90 \text{ mm}]$, respectively, and is also used to generate the zero-mean Gaussian noise $n^\tilde{y}_i$. The covariance matrix of $n^\tilde{y}_i$, $\Sigma^\tilde{y}_i$, is computed using (4.17). Then, data set $Y$ is generated by adding $n^\tilde{y}_i$ to each point in data set $\tilde{Y}$. Finally, the Umeyama algorithm and the UKF registration algorithm described in Section 4.1.2 are employed to register data set $\tilde{U}$ to $Y$. As mentioned earlier, the covariance
matrix of the state vector $\mathbf{x}$ is set by using (4.11), and assuming the rotation matrix $\mathbf{R}$ to be the identity matrix. Again, this procedure is repeated 10,000 times with different transformations and data sets generated randomly. To validate the UKF registration algorithm for cases where the additive Gaussian noise is anisotropic, the true covariance matrix $\Sigma^{y^i}$ is also used to register the two generated data sets.

Figure 4.5 shows the error histogram of the estimated translational and rotational parameters for the cases where the true and estimated values of the covariance matrix $\Sigma^{y^i}$ are utilized in the UKF registration algorithm. Also, the mean and standard deviation of the estimation error using the Umeyama and the UKF registration algorithms are given in Table 4.3. Table 4.3 and Figure 4.5 demonstrate that the performance of the UKF registration algorithm, using the estimation of $\Sigma^{y^i}$, equates to that of the UKF registration algorithm when employing the true value of $\Sigma^{y^i}$. Furthermore, in this simulation, the UKF and Umeyama registration algorithms estimate the registration parameters, with errors of less than 1 mm and 1°, 87 and 86 percent of the time, respectively. As can be seen from Table 4.3, the proposed algorithm is able to estimate the rotational parameters more accurately than the Umeyama algorithm. Both algorithms perform equally well when estimating the translational parameters. By assigning a different anisotropic zero-mean Gaussian noise distribution to each point of the data sets, the superiority of the proposed algorithm over the Umeyama, especially in the estimation of translational parameters, becomes more pronounced. To verify, we perform the same simulation as before with the difference that the points in the data sets are assumed to have different noise characteristics. It is assumed that one third, second third, and the rest of the points in both data sets are perturbed by zero-mean anisotropic zero-mean Gaussian noise with covariance matrices
(a) The UKF registration algorithm using the exact noise covariance matrix.

(b) The UKF registration algorithm using the estimated noise covariance matrix.

Figure 4.5: Error distribution of the estimated transformation parameters using the UKF registration algorithm for cases where (a) the covariance matrix of noise is known in advance; and (b) the covariance matrix of noise is estimated; over 5,000 trials in the presence of anisotropic zero-mean Gaussian noise.
Table 4.3: Mean and standard deviation of error for the estimated translational (mm) and rotational (degree) parameters using the Umeyama and UKF registration algorithms. The difference among the three techniques fails to show statistical significance for the estimated translation parameters ($p > 0.05$); however, for the estimated standard deviations of the rotation parameters, the Umeyama algorithm significantly differs from the other two methods ($p < 0.05$).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>UKF using $\Sigma^t_i$</th>
<th>UKF using $\hat{\Sigma}^t_i$</th>
<th>Umeyama Alg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>-0.003</td>
<td>-0.004</td>
<td>-0.004</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.005</td>
<td>0.004</td>
<td>0.005</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.004</td>
<td>-0.004</td>
<td>-0.004</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>-0.002</td>
<td>-0.002</td>
<td>-0.006</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>-0.0005</td>
<td>-0.0005</td>
<td>0.0004</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>0.00007</td>
<td>0.00004</td>
<td>0.001</td>
</tr>
</tbody>
</table>

$diag[2, 4, 6]$, $diag[6, 6, 1]$ and $diag[10, 1, 10]$, respectively.

As before, the mean and standard deviation of the estimation error for the Umeyama and the UKF registration algorithms (for both cases where the covariance matrix of the noise is known in advance and the covariance matrix of the noise is estimated during the registration process) are given in Table 4.4. The UKF and Umeyama registration algorithms estimate the registration parameters with an error of less than 1 mm and 1°, 88 percent and 82 percent of the time, respectively. Table 4.4 portrays how the proposed algorithm outperforms the Umeyama algorithm in the estimation of registration transformation parameters. It is clear that the proposed algorithm could significantly surpass the Umeyama algorithm when each point in
Table 4.4: Mean and standard deviation of error for the estimated translational (mm) and rotational (degree) parameters using the Umeyama and UKF registration algorithms (utilizing the exact and estimated values of the covariance matrix of anisotropic zero-mean Gaussian noise) where each one third portion of the data set points is perturbed by the same anisotropic zero-mean Gaussian noise distribution. The difference among the three techniques is statistically significant for the estimated standard deviations ($p < 0.05$).

<table>
<thead>
<tr>
<th>Estimation</th>
<th>Error</th>
<th>UKF using $\Sigma_i^y$</th>
<th>UKF using $\hat{\Sigma}_i^y$</th>
<th>Umeyama Alg.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>0.0006</td>
<td>0.37</td>
<td>0.00006</td>
<td>0.44</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.004</td>
<td>0.41</td>
<td>0.004</td>
<td>0.48</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.005</td>
<td>0.39</td>
<td>-0.004</td>
<td>0.46</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>-0.002</td>
<td>0.22</td>
<td>-0.003</td>
<td>0.27</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>-0.002</td>
<td>0.11</td>
<td>-0.001</td>
<td>0.14</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>0.001</td>
<td>0.22</td>
<td>0.0003</td>
<td>0.27</td>
</tr>
</tbody>
</table>

the data sets is perturbed by different anisotropic zero-mean Gaussian noise distribution. The reason being that, in contrast to the Umeyama algorithm, the UKF accounts for the variance of the different Gaussian distributions when estimating the registration transformation parameters. In addition, Table 4.4 demonstrates how the UKF registration algorithm using the exact value of $\Sigma_i^y$ slightly outperforms the UKF registration algorithm using the estimation of $\hat{\Sigma}_i^y$.

Finally, to verify the impact of replacing the rotation matrix $R$ with the identity matrix, we repeat the last two simulations by using the exact value of $R$ to calculate $\Sigma_i^\varepsilon$. As displayed in Tables 4.5 and 4.6, the performance of the proposed UKF registration algorithm is almost equal to the others reported in Tables 4.3 and 4.4, respectively.
Table 4.5: Mean and standard deviation of error of the estimated translational (mm) and rotational (degree) parameters using the UKF registration algorithms (utilizing the exact value of \( R \) to compute the covariance matrix of anisotropic zero-mean Gaussian noise) where each point in the data sets is perturbed by the same anisotropic zero-mean Gaussian noise distribution. The difference between the two cases fails to show statistical significance for all the reported parameters (\( p > 0.05 \)).

<table>
<thead>
<tr>
<th>Estimation</th>
<th>UKF using ( \Sigma_0^y )</th>
<th>UKF using ( \hat{\Sigma}_0^y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{\theta}_x - \theta_x )</td>
<td>-0.002</td>
<td>0.23</td>
</tr>
<tr>
<td>( \hat{\theta}_y - \theta_y )</td>
<td>-0.005</td>
<td>0.12</td>
</tr>
<tr>
<td>( \hat{\theta}_z - \theta_z )</td>
<td>0.0007</td>
<td>0.26</td>
</tr>
</tbody>
</table>

**d) Analyzing the sensitivity of the proposed algorithm to the point ordering:**

Data set \( \bar{U} \) is produced by drawing \( N = 30 \) points uniformly in the interval \([-250 \, mm, 250 \, mm]\). To create the noiseless fixed data set \( \bar{Y} \), a random transformation is applied to \( \bar{U} \). The transformation is generated by uniformly drawing the rotational and translational parameters in the intervals \([-90^\circ, 90^\circ]\) and \([-90 \, mm, 90 \, mm]\), respectively. An isotropic zero-mean Gaussian random noise vector \( \mathbf{n}_i^y \), with covariance of \( \Sigma_i^y = (1 \, mm^2) I_{3 \times 3} \), is added to each point in the data set \( \bar{Y} \) to generate the fixed data set \( Y \). Then, the proposed registration algorithm is used to register data set \( \bar{U} \) to \( Y \) for 5,000 trials. In each trial, the generated data sets are assumed to be fixed but the pairs of matching points are fed to the registration algorithm with different orders.
Table 4.6: Mean and standard deviation of error of the estimated translational (mm) and rotational (degree) parameters using the UKF registration algorithms (utilizing the exact value of $R$ to compute the covariance matrix of anisotropic zero-mean Gaussian noise) where each third portion of the data set points is perturbed by the same anisotropic zero-mean Gaussian noise distribution. The difference between the two cases is statistically significant for the reported standard deviations ($p < 0.05$).

<table>
<thead>
<tr>
<th>Estimation</th>
<th>UKF using $\Sigma^p_i$</th>
<th>UKF using $\hat{\Sigma}^p_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>-0.0006</td>
<td>0.37</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.004</td>
<td>0.41</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.005</td>
<td>0.39</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>-0.002</td>
<td>0.22</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>-0.002</td>
<td>0.11</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>0.001</td>
<td>0.22</td>
</tr>
</tbody>
</table>

The latter are generated randomly in each trial. Figure 4.6 graphs the convergence curve of the registration parameters for each trial. As can be seen, the algorithm has been able to converge to the true solution in all 5,000 trials. The graph verifies that the order of points is unimportant when the corresponding points between two data sets are known.

e) Analyzing the sensitivity of the proposed algorithm to the variance of FLE in observation model:

In this simulation, data set $\hat{\mathbf{U}}$ is produced by drawing $N = 30$ points uniformly in the interval $[-250 \text{ mm}, 250 \text{ mm}]$. To create the noiseless fixed data set $\hat{\mathbf{Y}}$, a random transformation is applied to $\hat{\mathbf{U}}$. The transformation is generated by uniformly
drawing the rotational and translational parameters in the intervals $[-90^\circ, 90^\circ]$ and $[-90 \text{ mm}, 90 \text{ mm}]$, respectively. An isotropic zero-mean Gaussian random noise vector $\mathbf{n}_i^{\hat{y}}$, with covariance of $\Sigma_i^{\hat{y}} = (1 \text{ mm}^2) \mathbf{I}_{3 \times 3}$, is added to each point in the data set $\hat{\mathbf{Y}}$ to generate the fixed data set $\mathbf{Y}$. Then, the proposed algorithm is used to register $\hat{\mathbf{U}}$ to $\mathbf{Y}$. In this experiment, it is assumed that knowledge of the variance of the observation model, $\Sigma_i^{\hat{y}}$, is imperfect and the proposed algorithm uses $\Sigma_i^{\hat{y}}$ with $\pm 20\%$ and $\pm 50\%$ error to register the data sets (see Tables 4.7 and 4.8). This procedure is repeated 5,000 times using different randomly generated transformations and data sets.

As shown in Tables 4.7 and 4.8, the performance of the proposed algorithm does not change even when there is 50% margin of error in the employed variance of the observation noise.
Table 4.7: Mean and standard deviation of error of the estimated translation (mm) and rotation (degree) parameters using the UKF registration algorithm when there is ±20% error in the variance of the observation noise. The difference between the two cases reported here and the ones estimated by the UKF registration algorithm in Table 4.1 fails to show statistical significance ($p > 0.05$).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>+20% error</th>
<th>-20% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$t_x - \hat{t}_x$</td>
<td>-0.004</td>
<td>0.18</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>0.002</td>
<td>0.18</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$</td>
<td>-0.003</td>
<td>0.18</td>
</tr>
<tr>
<td>$\theta_x - \hat{\theta}_x$</td>
<td>-0.0023</td>
<td>0.16</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>-0.0008</td>
<td>0.05</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>0.0002</td>
<td>0.16</td>
</tr>
</tbody>
</table>

It is interesting to note that the variance of the estimated parameters in Tables 4.7 and 4.8, over 5,000 trials, is identical for the two analyzed cases. This means that the estimation of the state vector is robust to the deviation of the variance of the observation noise from its nominal value (in the range of deviation shown). This behavior can be explained by looking at (4.8) and (4.11), where, in the case of isotropic noise, the covariance matrices of the observation and process model are proportional to each other. This leads to an almost constant Kalman gain (see Equation (3.7)), independent of the variance of the observation noise.

f) Analyzing the performance of the proposed algorithm using different number of observations:

In this experiment, a simulation similar to the previous one is run. The two data sets are generated as per the previous simulation, and the UKF registration algorithm is
Table 4.8: Mean and standard deviation of error of the estimated translation (mm) and rotation (degree) parameters using the UKF registration algorithm when there is ±50% error in the variance of the observation noise. The difference between the two cases reported here and the ones estimated by the UKF registration algorithm in Table 4.1 fails to show statistical significance (p > 0.05), except for the translation parameter along y axis, where the difference is significant (p = 0.013).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>+50% error</th>
<th>-50% error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$\hat{t}_x - t_x$</td>
<td>0.004</td>
<td>0.18</td>
</tr>
<tr>
<td>$\hat{t}_y - t_y$</td>
<td>-0.005</td>
<td>0.18</td>
</tr>
<tr>
<td>$\hat{t}_z - t_z$ (mm)</td>
<td>-0.002</td>
<td>0.18</td>
</tr>
<tr>
<td>$\hat{\theta}_x - \theta_x$</td>
<td>0.002</td>
<td>0.16</td>
</tr>
<tr>
<td>$\hat{\theta}_y - \theta_y$</td>
<td>-0.001</td>
<td>0.05</td>
</tr>
<tr>
<td>$\hat{\theta}_z - \theta_z$</td>
<td>-0.001</td>
<td>0.16</td>
</tr>
</tbody>
</table>

used to register the data sets 5,000 times for two scenarios. In the first scenario, the UKF registration algorithm uses the last and most recent observation to estimate the registration transformation parameters. In the second scenario, the last $i$ observations are employed to estimate the parameters.

Figures 4.7 and 4.8 display the estimated transformation parameters for both scenarios in a trial, respectively. As shown, both approaches estimated the transformation parameters accurately. However, the scenario which uses the last $i$ observation points more reliably estimates the variance of the transformation parameters, and its results closely match those obtained by numerical simulations. In fact, as Table 4.9 shows, not only are the variances yielded by the second approach less than those obtained by the first one, but they also tally with the results (generated by numerical simulations) listed in Table 4.1. In the next simulation, the performance of the proposed algorithm and EKF registration algorithm presented by Pennec and Thirion
Figure 4.7: Estimated registration transformation parameters using the last observation.

Figure 4.8: Estimated registration transformation parameters and their variances using the last $i$ observations.
CHAPTER 4. UKF PAIRWISE REGISTRATION

Table 4.9: Mean and standard deviation of error of the estimated translation (\textit{mm}) and rotation (\textit{degree}) parameters estimated by the UKF registration algorithm using the most recent and the last \( i \) observations, respectively. The difference between the standard deviations reported in the two cases is statistically significant (\( p < 0.05 \)).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>last observation ( i ) observations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>( t_x - t_x )</td>
<td>0.002</td>
</tr>
<tr>
<td>( t_y - t_y )</td>
<td>0.01</td>
</tr>
<tr>
<td>( t_z - t_z )</td>
<td>-0.004</td>
</tr>
<tr>
<td>( \theta_x - \theta_x )</td>
<td>0.001</td>
</tr>
<tr>
<td>( \theta_y - \theta_y )</td>
<td>-0.003</td>
</tr>
<tr>
<td>( \theta_z - \theta_z )</td>
<td>-0.009</td>
</tr>
</tbody>
</table>

\[98\] are compared.

g) Comparing the proposed algorithm with the EKF registration algorithm:

In this experiment, the proposed UKF registration algorithm is compared with the EKF registration algorithm proposed by Pennec and Thirion \[98\]. As before, the moving data set \( \bar{U} \) is generated by uniformly drawing 30 points in the interval \([-250 \text{ mm}, 250 \text{ mm}]\). The fixed data set \( Y \) is produced by applying a random transformation on the moving data set. The transformation is produced by uniformly drawing the rotational and translational parameters along all axes in the intervals \([-10^\circ, 10^\circ]\) and \([-10 \text{ mm}, 10 \text{ mm}]\), respectively. Also, isotropic zero-mean Gaussian noise with variance of 6 \text{ mm}^2 is added to the fixed data set. Then, the UKF and EKF registration algorithms are used to register the two data sets. This procedure is
repeated for 1,000 trials and, at each trial, as previously explained, the transformation and data sets are randomly generated.

Tables 4.10 and 4.11 show the error of the estimated transformation parameters using both algorithms for two different simulations, where $N$ (number of the points in the data sets) is chosen to be 30 and 150, respectively. From the Tables, one can conclude that the UKF registration algorithm needs fewer points than EKF in order to converge to the true solution. The performances of UKF and EKF for the translational parameters are identical since those parameters are the linear terms in the observation model. However, the UKF registration algorithm outperforms the EKF since, as mentioned earlier, the former considers higher orders of the Taylor series expansion of the nonlinearity. In fact, even when it increases the number of registration points from 30 to 150, the EKF registration algorithm still cannot attain

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>UKF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Std.</td>
</tr>
<tr>
<td>$t_x - t_x$</td>
<td>0.01</td>
<td>0.44</td>
</tr>
<tr>
<td>$t_y - t_y$</td>
<td>0.001</td>
<td>0.44</td>
</tr>
<tr>
<td>$t_z - t_z$</td>
<td>-0.02</td>
<td>0.44</td>
</tr>
<tr>
<td>$\theta_x - \theta_x$</td>
<td>-0.005</td>
<td>0.14</td>
</tr>
<tr>
<td>$\theta_y - \theta_y$</td>
<td>-0.0001</td>
<td>0.14</td>
</tr>
<tr>
<td>$\theta_z - \theta_z$</td>
<td>0.002</td>
<td>0.14</td>
</tr>
</tbody>
</table>
Table 4.11: Mean and standard deviation of error of the estimated translation \((\text{mm})\) and rotation \((\text{degree})\) parameters estimated by the UKF and EKF registration algorithms when \(N = 150\). The difference in the two reported cases fails to show statistical significance for translation parameters \((p = 1)\); however, the difference between the reported standard deviations is statistically significant for the rotation parameters \((p < 0.05)\).

<table>
<thead>
<tr>
<th>Estimation Error</th>
<th>UKF</th>
<th>EKF</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\hat{t}_x - t_x)</td>
<td>-0.01 0.2</td>
<td>-0.01 0.2</td>
</tr>
<tr>
<td>(\hat{t}_y - t_y)</td>
<td>0.004 0.2</td>
<td>0.004 0.2</td>
</tr>
<tr>
<td>(\hat{t}_z - t_z)</td>
<td>-0.01 0.2</td>
<td>-0.01 0.2</td>
</tr>
<tr>
<td>(\hat{\theta}_x - \theta_x)</td>
<td>0.003 0.06</td>
<td>-0.009 0.45</td>
</tr>
<tr>
<td>(\hat{\theta}_y - \theta_y)</td>
<td>0.001 0.06</td>
<td>0.01 0.45</td>
</tr>
<tr>
<td>(\hat{\theta}_z - \theta_z)</td>
<td>0.003 0.06</td>
<td>-0.02 0.45</td>
</tr>
</tbody>
</table>

the performance of UKF. We have observed that the EKF registration algorithm requires more than 200 points to achieve the same performance as that of UKF using 30 registration points. The result may have direct practical implications, specifically in computer-aided surgery, when only limited number of anatomical points can be captured from the surgical site.

### 4.2 UKF Pairwise Surface-based Registration

Here, we will extend the proposed UKF registration algorithm to register two surfaces (data sets) where the corresponding pair matches between the data sets are unknown. Each data set may differ in size and, therefore, only partially overlap the other set. In such a case, a well-known method called the Iterative Closest Points (ICP) algorithm [11] can register these data sets.

Let \(U\) be the moving data set and \(Y\) be the fixed data set to be registered. Both
sets are generated from a rigid-body surface and decomposed into three-dimensional point set forms. Let $N_U$ and $N_Y$ represent the number of points in data sets $U$ and $Y$, respectively. In this case, we assume that the corresponding points in $U$ and $Y$ are unknown.

The ICP algorithm estimates the rotation matrix $R$ and translation vector $t$ by minimizing the cost function $C$ in (4.2). In this minimization, $y_i$, the corresponding point of $u_i$, is estimated as

$$y_i = f_{ICP}(Y, Ru_i + t), \quad (4.24)$$

where $f_{ICP}$ is the operator that finds the closest point in the data set $Y$ to $(Ru_i + t)$. By estimating the corresponding matches between the two data sets, $R$ and $t$ can be determined using one of the algorithms proposed by Arun [2], Horn [47, 48] or Umeyama [126]. Then, the moving data set $U$ is transformed under the estimated transformation parameters. The procedure iterates until it converges to a locally optimum solution or, until the surface registration error between two data sets reduces to a certain threshold value. At each iteration, new correspondences between the two data sets are established by using the updated transformation parameters from the previous iterations.

One of the drawbacks of the ICP algorithm is its sensitivity to the initial alignment. If the sets are not closely aligned, the closest point operator in the ICP algorithm returns incorrect matches, which can easily cause the ICP algorithm to become trapped in a local minimum.

The UKF registration algorithm can alleviate the ICP limitations by incorporating the estimated variances of the transformation parameters, and covariance matrices of the observation model and process model into the registration process.
There exist two sources of noise in the observation model: 1) the noise emanating from the fiducial (point) localization error in the data sets (extrinsic error), and 2) the noise caused by false matches (intrinsic error) [107]. The errors for the $i$th point are modelled by covariance matrix of the observation model, $\Sigma_i$, as

$$\Sigma_i = \Sigma_{i}^{\hat{y}} + \Sigma_{i}^{int},$$  \hspace{1cm} (4.25)

where $\Sigma_{i}^{\hat{y}}$ models the fiducial localization error in the data sets, and $\Sigma_{i}^{int}$ models the false matches. $\Sigma_{i}^{\hat{y}}$ depends on the variance of the noise in the data and is constant during the registration. However, $\Sigma_{i}^{int}$ changes during the registration process.

At the beginning of the registration where the point matches, reported by the closest point algorithm, are most likely incorrect, $\Sigma_{i}^{int}$ is large and therefore the UKF registration algorithm does not update the state vector based on the matches. As the algorithm converges to a solution, $\Sigma_{i}^{int}$ is reduced to let the state vector be tuned. $\Sigma_{i}^{int}$, in our simulations, is assumed to be fixed for every point in the data sets. Specifically, it is set to be the mean square distance of the matching points reported by the closest point operator, as proposed in [107] based on the results of Sharp et al. [111].

Initially, when the distance between two data sets is large, $\Sigma_{i}^{int}$ is large as well. As the data sets near each other, $\Sigma_{i}^{int}$ reduces. First, the covariance matrix of the process model, $\Sigma_{i}^{x}$, is calculated using the initial uncertainty of each transformation parameter. This uncertainty possibly allows us to move from a poor initial estimation of the state to a better one. It is also assumed that $\Sigma_{i}^{x}$ is fixed for every point in the data sets and that it is annealed over time to a small value. We annealed $\Sigma_{i}^{x}$ by a factor of 0.95 after each iteration. The UKF registration algorithm works in a similar fashion to the ICP algorithm with two major differences: it incrementally processes
In what follows, the proposed surface-based registration algorithm is explained.

4.2.1 Isotropic and Anisotropic FLE Distributions

Here, we propose UKF surface-based registration algorithm to register two surfaces in the presence of isotropic and anisotropic zero-mean Gaussian noise. To finely tune the registration between two surfaces that are roughly aligned, the following algorithm, as shown in Figure 4.9, iterates until it converges to a solution:

1) Initialize the state vector $\mathbf{x}$ and its covariance matrix with zero and the identity matrix, respectively. Set the covariance matrix of the observation model, $\Sigma_i$, as explained in (4.25). Also, set the covariance matrix of the process model for every point

Figure 4.9: Flowchart of UKF surface-based registration algorithm.

the points in the data sets, and uses the variance information to deal with outliers.
in the data sets, $\Sigma^x_i$, to the initial uncertainty for each transformation parameter.

2) Append the $i$th point from moving data set $U$ to the previously selected points from that data set. These points are randomly selected from the moving data set.

3) Find the corresponding points between the selected points from the moving data set $U$ and fixed data set $Y$ by using the closest points and the k-d tree algorithms [9], and determine the mean square distance error, $E[d^2]$, among the estimated point matches, as

$$Y_{1:i} = f_{CIP}(Y, U_{1:i}), \quad (4.26)$$

$$E[d^2] = \frac{1}{i} \sum_{k=1}^{i} \|y_k - u_k\|^2. \quad (4.27)$$

4) If FLE has an isotropic distribution, then use the proposed algorithms in Section 4.1.1 and the estimated corresponding points to compute the rotational and translational parameters. If FLE has a anisotropic distribution, use the proposed algorithm in Section 4.1.2 and the computed corresponding points to derive the transformation parameters.

5) Use the estimated transformation parameters to update the moving data set $U$, as

$$u_i = R(\hat{x}_i) u_i + \hat{x}_t, \quad i = 1, \ldots, N. \quad (4.28)$$

6) If $E[d^2]$ is less than a certain threshold, stop the algorithm; otherwise, update the covariance matrix of the observation model with $\Sigma_i = \Sigma^y_i + E[d^2]I$. Also, anneal $\Sigma^x_i$ with the factor of 0.95 and go to Step 2.

This algorithm iterates through all the points in $U$ to converge to a solution.

The UKF registration algorithm needs initial rotational and translational parameters to start the registration with. If these values strongly differ from the true solution,
the algorithm may not converge to the optimum solution.

The complexity of the proposed registration algorithm is nearly the summation of
the complexity of the k-d tree algorithm for finding the matching points between the
two data sets ($\mathcal{O}(N_U \log N_Y)$) [9], together with the complexity of UKF for estimating
the transformation parameters which are governed by the complexity of root square
matrix computation [80], ($\mathcal{O}(N_U^3)$), as

$$
\mathcal{O}_{UKF} = \mathcal{O}(N_U^3) + \mathcal{O}(N_U \log N_Y). 
$$

However, it should be noted that the complexity of finding the square root of a matrix
can be reduced to $\mathcal{O}(N_U^2)$ [80] using some approximations that are not used in our
implementation.

The reported complexity in (4.29) is relative to the number of points in the moving
data set, $N_U$, and shows the number of multiplications in the algorithm. Since the
number of points in the proposed algorithm varies at each iteration from 1 to $N_U$ (the
points are incrementally fed to the proposed algorithm), one could conclude that, on
average, the complexity of the algorithm is less than (4.29).

Here, it is fair to compare the complexity of the proposed algorithm with that of
the standard ICP algorithm. The complexity of the latter is the summation of the
complexity of the k-d tree algorithm, ($\mathcal{O}(N_U \log N_Y)$), and the complexity of finding
transformation parameters, $\mathcal{O}(N_U)$, as

$$
\mathcal{O}_{ICP} = \mathcal{O}(N_U) + \mathcal{O}(N_U \log N_Y). 
$$

In contrast to the UKF registration algorithm, this complexity is fixed during each
iteration since the ICP algorithm uses all the points in the data sets at each iteration
to perform registration.
Incremental processing of the points in data sets, in the UKF registration, enables us to stop the registration procedure when the desired registration accuracy is achieved. In addition, incremental processing of points allows the algorithm to be executed while the registration points are incrementally collected from the object. Rather than collecting all the points first, and then running the registration algorithm, the proposed algorithm starts registering the data sets by feeding the points one by one to the UKF algorithm while they are collected from the object. Although the ICP registration algorithm can be modified to incrementally process the data sets and use the same subset of points per iteration as does the UKF registration algorithm, the experimental results in the next section will show that this modification does not improve the ICP registration algorithm’s performance.

In the next section, based on our performed simulations and experimental results, we show that the proposed registration algorithm is more robust than the standard ICP algorithm, and is appropriate for a wider range of initial alignments.

4.2.2 Numerical Simulations

In this section, the reliability of our algorithm has been tested and compared with that of the ICP algorithm on different data sets, without knowing the correspondences. In the first simulation, the data are randomly generated from uniform distributions. However, in the second and third simulations, the proposed algorithm is verified on real data sets sampled randomly from a pelvic cadaver and a scaphoid bone phantom, respectively. The performance of the UKF and ICP registration algorithms in registering these data sets are subsequently compared.
a) Registering randomly generated data sets:

In this simulation, the moving data set $U$ is randomly generated by drawing 2,500 three-dimensional points uniformly in the interval $[-250 \text{ mm}, 250 \text{ mm}]$. A random transformation within the range of $[\pm 10^\circ]$ and $[\pm 10 \text{ mm}]$ is applied to the moving data set, and isotropic zero-mean Gaussian random noise with a variance of $3 \text{ mm}^2$ is added to produce the fixed data set $Y$. Then, the ICP and UKF registration algorithms are employed to register the first randomly selected 350 points of $U$ to all the points of $Y$. This procedure is repeated 500 times. In each trial, a different random transformation is applied to the moving data set, and the UKF and ICP registration algorithms are stopped when the root mean square of point registration error decreases to a certain threshold level ($1 \text{ mm}$). The registration error incurred by the UKF and ICP registration algorithms over 500 trials for the remainder of the points not used in the registration is given in Figure 4.10. As shown in Figure 4.10, the UKF registration algorithm is able to register the data sets more accurately than the ICP registration algorithm. On the other hand, the ICP registration algorithm becomes trapped in local minima in many trials. In 500 trials, ICP converged to the true solution 67 percent of the time. However, the UKF registration algorithm did not fail in any of the 500 trials. The error that is observed in the UKF registration error histogram (Figure 4.10-a) is caused by the Gaussian noise added to the fixed data set. If the variance of the noise falls to zero, this error vanishes as well.

Figure 4.11 depicts the processing time of the UKF and ICP registration algorithms. As expected, the running time of the UKF registration algorithm exceeds that of the ICP registration algorithm due to its higher computational complexity. In some cases, the UKF registration algorithm takes more than 20 seconds to find the
Figure 4.10: Comparing the registration error caused by the UKF registration and standard ICP registration algorithms for registering two randomly generated data sets over 500 trials in the presence of isotropic zero-mean Gaussian noise.
Figure 4.11: Comparing the processing time histograms of the UKF and standard ICP registration algorithms to register two randomly generated data sets over 500 trials in the presence of isotropic zero-mean Gaussian noise. Note that the time scales are different.
solution. This happens when the algorithm needs more points for convergence. However, on average, the UKF registration algorithm requires fewer points than does the ICP algorithm to perform the registration. At the beginning of the registration, it is not known how many registration points should be considered for the ICP algorithm to converge. But, since the UKF registration algorithm incrementally processes the points, it can stop processing points when data sets are registered within the desired accuracy.

![Histogram](image)

Figure 4.12: Histogram of the number of points used by the UKF registration algorithm to register two randomly generated data sets over 500 trials. The UKF registration algorithm stops considering points when the root mean square of point registration error decreases to a certain threshold level (1 mm). Number of points in the moving and fixed data sets are 350.

Figure 4.12 shows the histogram of the number of points used by the UKF registration algorithm to register the two data sets over 500 trials. We note that the minimum number of points required by the UKF registration algorithm is 4; however, the maximum number of points depends on the initial alignment of the two data sets and on the variance of the noise added to the fixed data set.
We also modified the ICP registration algorithm to use the same subset of points utilized by the UKF registration algorithm to register the generated data sets. Figure 4.13 displays the registration error histogram computed from the points not used in the registration process when employing the modified ICP registration algorithm. As it is shown, the modified ICP registration algorithm fails to accurately register the data sets in 500 trials. This illustrates that incrementally processing the points does not necessarily improve the ICP registration algorithm’s performance.

We also performed the same simulation as before but used larger random transformations to determine where the UKF registration and standard ICP registration algorithms break down. The two data sets are generated as before, but the transformations are produced by randomly choosing the translational and rotational parameters within cubes with sides of [±1 mm, ±1°], [±5 mm, ±5°], [±10 mm, ±10°], [±20 mm, ±20°], [±30 mm, ±30°] and [±40 mm, ±40°], respectively. The rate of
Table 4.12: Convergence rate of the UKF, ICP and incremental ICP algorithms for different ranges of random transformations.

<table>
<thead>
<tr>
<th>Range of Transformation</th>
<th>UKF Algorithm</th>
<th>ICP Algorithm</th>
<th>Incremental ICP Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>[±1 mm, ±1°]</td>
<td>100%</td>
<td>82%</td>
<td>91%</td>
</tr>
<tr>
<td>[±5 mm, ±5°]</td>
<td>100%</td>
<td>72%</td>
<td>62%</td>
</tr>
<tr>
<td>[±10 mm, ±10°]</td>
<td>100%</td>
<td>67%</td>
<td>57%</td>
</tr>
<tr>
<td>[±20 mm, ±20°]</td>
<td>100%</td>
<td>56%</td>
<td>50%</td>
</tr>
<tr>
<td>[±30 mm, ±30°]</td>
<td>97%</td>
<td>54%</td>
<td>57%</td>
</tr>
<tr>
<td>[±40 mm, ±40°]</td>
<td>68%</td>
<td>45%</td>
<td>40%</td>
</tr>
</tbody>
</table>

convergence for both algorithms over 100 trials is listed in Table 4.12.

As can be seen, the ICP algorithm cannot register the two data sets in all trials even when the range of random transformations remains small. In contrast, the UKF registration algorithm perfectly registers the two data sets in all trials for the small range of random transformations, and its performance starts declining for random transformations larger than 20 mm and 20°. This experiment illustrates that the UKF registration algorithm is much more robust than the ICP registration algorithm to the initial alignments.

In the last experiment in this section, we register two randomly generated data sets, in the same range as before, in the presence of anisotropic zero-mean Gaussian noise in the moving and fixed data sets with the following covariance matrices

\[ \Sigma^u_i = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix}, \quad \Sigma^y_i = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix}. \]

In this simulation, the rotational and translational parameters are randomly drawn
in the intervals $[-20^\circ, 20^\circ]$ and $[-20 \text{ mm}, 20 \text{ mm}]$, respectively. Also, the UKF and ICP registration algorithms are stopped once the root mean square of the point registration error decreases to 2 mm. Figure 4.14 demonstrates the histogram of the registration error, generated by UKF and ICP registration algorithms over 100 trials for the points not used in the registration. The processing time of the UKF and ICP registration algorithms are given in Figure 4.15. Again, the UKF registration algorithm registers the generated data sets more accurately and robustly than the ICP algorithm.

Figure 4.16 displays the histogram of the number of points used by the UKF registration algorithm over all trials.

b) Registering a randomly selected point set to the CT images of a pelvic bone:

Here, the fixed data set $Y$ is generated by taking CT images of a pelvic cadaver bone. The resulting CT images are then segmented manually using the commercial software Amira (TGS, Berlin, Germany) to produce a three-dimensional mesh model containing 2,500 points. The moving data set $U$ is generated by randomly choosing 200 points from the CT mesh data and applying a random transformation on the collected points. The random transformation is generated by uniformly drawing the translational and rotational parameters in the intervals $[-20^\circ, 20^\circ]$ and $[-20 \text{ mm}, 20 \text{ mm}]$, respectively. In this simulation, no noise is added to the generated data sets to compare the performance of the UKF and ICP registration algorithms under conditions when data sets are not perturbed by noise. The UKF and ICP registration algorithms are used to register the first randomly selected 100 points of
CHAPTER 4. UKF PAIRWISE REGISTRATION

Figure 4.14: Comparing the registration error caused by the UKF and standard ICP registration algorithms when registering two randomly generated data sets over 100 trials in the presence of anisotropic zero-mean Gaussian noise.
Figure 4.15: Comparing the processing time histograms of the UKF and standard ICP registration algorithms to register two randomly generated data sets over 100 trials in the presence of anisotropic zero-mean Gaussian noise.
Figure 4.16: Histogram of the number of points used by the UKF registration algorithm to register two randomly generated data sets over 100 trials in the presence of anisotropic zero-mean Gaussian noise.

the moving data set $\mathbf{U}$ to the CT mesh data set $\mathbf{Y}$. This procedure is repeated 50 times.

Figure 4.17 shows how the moving data set $\mathbf{U}$ is registered to the CT mesh by the UKF registration algorithm. Figure 4.18 is the histogram of the root mean square registration error (for the points not used in the registration) that are prompted by the UKF and ICP registration algorithms over 50 trials. As illustrated, the maximum point registration error for the UKF registration algorithm is less than 0.025 mm. However, the presence of local minima causes this error to increase to 1.5 mm for the ICP registration algorithm.

In this subsection, another simulation is performed on the same data sets to examine the robustness of the proposed algorithm to the outliers. Here, the outliers are defined as the points in the first data set which find no matches in the second one. As before, moving data set $\mathbf{U}$ is generated by randomly choosing 200 points from
the CT mesh data and applying a random transformation on the collected points. The random transformation is generated by uniformly drawing the translational and rotational parameters in the intervals $[-20^\circ, 20^\circ]$ and $[-20 \text{ mm}, 20 \text{ mm}]$, respectively. To simulate the outliers, some percent of the first 100 points of the moving data set are randomly selected and perturbed by a zero-mean Gaussian noise with variance of $1 \text{ mm}^2$. Then, the ICP and UKF registration algorithms are used to register the data sets. This procedure is repeated 100 times by applying random transformations and adding zero-mean Gaussian noise to 10% and 20% of the points in the moving data set, respectively.

In this simulation, it is assumed that the ICP and UKF registration algorithms have converged to the true solution when the mean distance error is less than 2 mm. Table 4.13 presents the mean distance error of the points not used in the registration.
Figure 4.18: Comparing the histograms of the root mean square of the registration errors resulted from the UKF and ICP registration algorithms (for registering the CT mesh model of the pelvic cadaver bone to a set of randomly generated points from that CT mesh model) over 50 registration trials. Note that the time scales are different.
Table 4.13: Mean square distance error ($mm^2$) and convergence rate of the ICP and UKF registration algorithms in the presence of outliers over 100 trials. Outliers are generated by a zero-mean Gaussian noise with variance of 1 $mm^2$.

<table>
<thead>
<tr>
<th>Percentage of outliers</th>
<th>UKF algorithm</th>
<th>ICP algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Convergence rate</td>
<td>Error</td>
</tr>
<tr>
<td>10%</td>
<td>100%</td>
<td>0.024</td>
</tr>
<tr>
<td>20%</td>
<td>100%</td>
<td>0.045</td>
</tr>
</tbody>
</table>

between the two data sets for different cases. As can be seen, the proposed UKF registration algorithm proves to be more robust than the ICP registration algorithm in registering the two data sets in the presence of outliers. The mean square distance error of the ICP registration algorithm is substantial since the ICP registration algorithm is trapped in local minima in most of the cases; however, the UKF registration algorithm is able to converge to the true solution in all the trials. Also, by increasing the number of outliers, the convergence rate of the ICP algorithm decreases. But this is not the case for the proposed UKF registration algorithm.

c) Registration of ultrasound and CT images of a scaphoid bone phantom:

In this simulation, the fixed data set $Y$ is generated by taking CT images of a scaphoid bone phantom. CT images are taken using a LightSpeed Plus CT scanner (GE Medical Systems, Waukesha WI). Then, Mesher software developed at our institution is used to semi-automatically segment the geometry of the scaphoid bone phantom from the CT images, and to create a three-dimensional surface model –that contains 17,029 points– using a marching cubes algorithm. The generated three-dimensional surface model is shown in Figure 4.19.
To produce the moving data set $U$, a set of tracked free-hand ultrasound images (30 images) is captured from the bone phantom submerged in a water-bath. Figure 4.20 portrays one of the ultrasound images taken from the surface of the scaphoid bone phantom.

Using the method proposed in [15] and [99], these images are segmented and calibrated, and then 450 three-dimensional ultrasound points are generated which represent the moving data set $U$, as shown in Figure 4.21. The coordinate frame of CT mesh data is very closely aligned to the coordinate frame of the three-dimensional ultrasound points. This alignment is confirmed by observing the locations of the fiducial markers mounted on the scaphoid bone phantom in the ultrasound and CT coordinate frames. After alignment, the closest points between the ultrasound and CT data sets are determined. These points are then used as the ground truth to calculate the registration error. Next, 50 random transformations are applied to the
three-dimensional ultrasound data to steer it away from the CT mesh. Finally, the
UKF and ICP registration algorithms are used to fit the three-dimensional ultrasound
points back to the CT mesh points. Each random transformation is generated by
uniformly drawing the three rotational and three translational parameters in the
intervals \([-10^\circ, 10^\circ]\) and \([-10 \text{ mm}, 10 \text{ mm}]\), respectively. Figure 4.22 illustrates the
three-dimensional ultrasound points that are registered to the CT data set by using
the UKF registration algorithm.

Figure 4.23 gives the registration error histogram (the distance between the regis-
tered three-dimensional ultrasound points and the CT points defined as ground truth)
created by the UKF and ICP registration algorithms over 50 trials. It is noted that
the mean registration error for the UKF registration algorithm is 0.27 mm; however,
this error increases to 2.37 mm for the ICP registration algorithm due to the local
minima.
CHAPTER 4. UKF PAIRWISE REGISTRATION

Figure 4.21: Generated three-dimensional ultrasound points from the scaphoid bone phantom.

Figure 4.22: Registration of the three-dimensional ultrasound data set to the CT mesh model of the scaphoid bone phantom using the UKF registration algorithm. The units in the figure are in mm.
CHAPTER 4. UKF PAIRWISE REGISTRATION

Figure 4.23: Comparing the histograms of the registration errors that result from the UKF and ICP registration algorithms (for registering CT mesh model of the scaphoid bone phantom to the collected three-dimensional ultrasound points from the same phantom) over 50 registration trials. Note that the time scales are different.
4.3 Summary and Discussion

A new point-based, rigid-body registration algorithm, called the UKF registration algorithm, has been presented. This algorithm is based on the UKF algorithm. As discussed earlier, the UKF algorithm is a powerful technique which can estimate a state vector in a nonlinear system (which is corrupted by Gaussian noise) with the accuracy of at least the second-order Taylor series expansion. An extensive set of experiments demonstrating the algorithm’s performance under various noise, outlier and initial alignment conditions, for a large range of data sets, is performed. The summary and discussion of these experiments are given below.

First, the UKF registration algorithm is tested on two randomly generated data sets (with known corresponding points) in the presence of isotropic zero-mean Gaussian noise. The simulation results evidence that, as shown in Table 4.1, the performance of the UKF registration algorithm equates that of the algorithms proposed by Umeyama [126] and Horn et al. [48] when FLE has an identical isotropic zero-mean Gaussian distributions.

The crucial property of the proposed algorithm is that, as depicted in Figure 4.4, it can compute the variance of the registration parameters at the same time as performing the registration. Table 4.1 and Figure 4.4 illustrate that the variances, which are derived by the proposed algorithm in only one registration trial, closely match the ones derived by the Umeyama algorithm over 5,000 registration trials. The calculation of variances may potentially be utilized for further applications such as the estimation of Target Registration Error (TRE) at a desired target location [36, 83].

Table 4.2 shows the shortcoming of the Umeyama algorithm compared to the
proposed technique where each point in the data set is perturbed by a different isotropic zero-mean Gaussian noise distribution. The proposed approach outperforms the Umeyama algorithm in this scenario, since it can consider different weights (variances) for each point in the data sets; however, the Umeyama algorithm assumes all the points have the same weight.

The sensitivity of the proposed algorithm to the variance of the noise in the observation model is then examined. Tables 4.7 and 4.8 show that the proposed algorithm is not sensitive to the variance of the noise in the observation model up to the ±50% changes of the variance of the observation noise from its nominal value. However, we have to emphasize that, although the calculation of registration parameters is not overly affected by the variance of the noise in the observation model, the computed variances of the registration parameters are sensitive to the variance of the observation noise. In other words, by changing the variance of the noise in the observation model from its nominal value, the estimated registration parameters remain the same; however, their computed variances, which are related to the variance of the noise in the observation model, no longer remain accurate.

The UKF registration algorithm is then extended to the case where the noise added to the data sets is anisotropic zero-mean Gaussian noise. In this case, the modified covariance matrix \( \Sigma_y \) depends on the unestimated rotational parameters. The UKF registration algorithm is modified to register the two data sets, and meanwhile, to estimate the modified covariance matrix. By means of numerical simulations as shown in Tables 4.3 and 4.4, it is verified that the performance of this algorithm nearly matches that of the ideal case where the true value of the modified covariance matrix is known in advance. It also outperforms the Umeyama algorithm since, as mentioned
earlier, the proposed algorithm can assign different weights for each point (along each axis) in the registration process.

The proposed algorithm is also compared with its peer, EKF registration algorithm. Tables 4.10 and 4.11 verify that the proposed algorithm is more reliable than the EKF registration algorithm in registering two data sets. In addition, the proposed technique needs fewer points than the EKF registration algorithm to compute the registration parameters.

In the next step, the UKF registration algorithm is extended to the case where the corresponding points between data sets are unknown. The developed algorithm is compared with the well-known Iterative Closest Point (ICP) algorithm by applying both methods to register different point sets. The data sets are either generated randomly from a uniform distribution or garnered from a pelvic cadaver and a scaphoid bone phantom. First, the UKF and ICP registration algorithms are employed to register two randomly generated data sets, with the same number of points, in the presence of Gaussian noise. Figures 4.10 and 4.14 illustrate that the UKF registration algorithm is more accurate than the ICP registration algorithm in registering two data sets perturbed by Gaussian noise. Table 4.12 displays the superior robustness of the proposed algorithm compared to the ICP algorithm in registering two data sets for different ranges of transformation.

In the next experiments, the real data sets are used to examine the performance of the proposed algorithm in real clinical applications. The first data set is a 3D mesh model generated from CT images of a pelvic cadaver bone. Figure 4.18 shows the advantage of the proposed algorithm over the ICP registration algorithm in registering the two data sets. Furthermore, Table 4.13 displays how the proposed algorithm
is more robust than the ICP registration algorithm to the outliers in registering the generated data sets. In the last experiment, the 3D mesh model, generated from CT images of a scaphoid bone phantom, is registered to the three-dimensional ultrasound points (collected from the same phantom), using the UKF and the ICP registration algorithms. In this simulation, the data sets have different sizes of points and are perturbed by segmentation and calibration errors. As before, Figure 4.23 demonstrates that the UKF registration algorithm is more accurate and accepts a wider range of initial alignments in comparison to the ICP registration algorithm.

Based on these results, it is evident that the UKF registration algorithm could potentially be used as an alternative for applications to which traditionally, the ICP registration technique has been applied. The proposed algorithm is also a good candidate for situations where data sets have large percentage of outliers, and where accurate initial alignment for registration may not be possible or would be time consuming (outliers and large initial alignment errors would normally cause the ICP algorithm to fail). The proposed approach also enables the initialization of the registration process, while the registering data is collected (e.g., by using a stylus pointer during a surgical operation). Furthermore, since the proposed technique also provides an estimate of the calculated variances for the registration parameters, it could be used for applications where monitoring the registration accuracy (e.g., before initiating a surgical procedure) is of vital importance.

Since the initial publication of the proposed algorithm [81], the proposed method has been used in a computer-assisted surgery experiments and in ultrasound calibration. Armand et al. [1] developed a computer-aided navigation system, which uses the proposed UKF registration algorithm to align 3D models of pelvises with their
corresponding cadavers. The registration algorithm was shown to be sufficiently accurate for such application. The proposed algorithm has also been integrated into a surgical navigation system at Harvard Medical school which requires registration of MRI images to patients [5]. Furthermore, the proposed algorithm was extended to calculate free-hand ultrasound image calibration parameters [88]. In this case, state vector $x$ includes 3 rotational parameters, 3 translational parameters, and 3 scaling parameters that are estimated by the UKF algorithm. The results of calibration experiments were very promising and demonstrated the potential of the technique for non-rigid registration applications.
Chapter 5

UKF Multi-body Registration

In this chapter, we extend the proposed pairwise UKF registration algorithms in Chapter 4 to simultaneously align multiple data sets. First, we introduce the UKF multi-body point-based registration algorithm to simultaneously align multiple sets of corresponding points in the presence of FLE with an arbitrary zero-mean Gaussian distribution. Then, the proposed algorithm is extended to the multi-body surface-based registration algorithm that concurrently aligns multiple surfaces without knowledge about the corresponding points among the surfaces. Lastly, the proposed algorithm is used to solve a clinical problem in computer-assisted orthopedic surgeries where multiple fracture surfaces of a bone should be registered back to their correct healing positions. We introduce a potential solution to this problem based on our proposed multi-body surface based registration algorithm. The main achievements of this chapter are:

1) The proposed UKF multi-body point-based registration algorithm outperforms the standard multi-body point-based registration algorithm in the literature proposed by Pennec [97].
2) The proposed UKF multi-body surface-based registration algorithm can accurately and simultaneously register multiple surfaces. Corresponding points among surfaces are not known, but surfaces are roughly aligned such that the closest points among the surfaces are a reliable estimate of the corresponding points.

3) The potential solution to the clinical orthopedic problem can successfully register multiple bone fractures back to their correct healing positions without imposing conditions on preliminarily alignment of the bone fractures.

5.1 UKF Multi-body Point-based Registration

In this section, we propose to use the Unscented Kalman Filter (UKF) algorithm to simultaneously register multiple point sets when the corresponding points among the data sets are known and FLE has an isotropic zero-mean Gaussian distribution. Then, by means of numerical simulations, the performance of the proposed algorithm is compared with the well-known multiple point set registration algorithm in the literature, which was presented by Pennec [97].

5.1.1 Isotropic FLE Distribution

Let’s assume there are $M$ corresponding point sets ($S^1, S^2, \ldots, S^M$), and we would like to simultaneously estimate $M - 1$ rigid transformations ($T_2, T_3, \ldots, T_M$) that register the points sets to each other in the presence of isotropic zero-mean Gaussian FLE. Data set one is assumed to be in the reference frame ($T_1 = I_{3 \times 3}$).

Each rigid transformation $T_\alpha$ consists of a rotation matrix $R_\alpha(\theta_\alpha^x, \theta_\alpha^y, \theta_\alpha^z)$, and a translation vector $t_\alpha = [t_\alpha^x, t_\alpha^y, t_\alpha^z]$. Therefore, $(M - 1) \times 6$ parameters are needed to
estimate $M - 1$ rigid transformations. We define the state vector $x$ comprising these transformation parameters as

$$x = [t^2_x, t^2_y, t^2_z, \theta^2_x, \theta^2_y, \theta^2_z, ..., t^M_x, t^M_y, t^M_z, \theta^M_x, \theta^M_y, \theta^M_z]^T,$$

(5.1)

$$= [x^2_T, x^2_T, ..., x^MT, x^MT]^T,$$

where $x^\alpha_T = [t^\alpha_x, t^\alpha_y, t^\alpha_z]^T$ and $x^\alpha_T = [\theta^\alpha_x, \theta^\alpha_y, \theta^\alpha_z]^T$. As before, the process model is assumed to be governed by the linear function as follows:

$$x_k = x_{k-1} + N(0, \Sigma_Q),$$

(5.2)

where $x_k$ is the defined state vector at time $k$ (when the $k$th pair of corresponding points is fed into the algorithm) with the initial value and covariance matrix $x_0$ and $P_0$, respectively. $N(0, \Sigma_Q)$ is a zero-mean, Gaussian random vector with covariance matrix $\Sigma_Q$.

The state vector $x_k$ must be incrementally estimated from the observation model, which is a matrix of functions defined as follows:

$$y_{1:k} = \begin{cases} p_{1:1}^{1,\beta} = R_{(x^\beta_0)} p_{1:1}^{1,1} + \mathbf{t}(x^\beta_0) + N(0, \Sigma_{1:1}^{1,\beta}), \alpha = 1, \\ 0 = R_{(x^\beta_0)} p_{1:1}^{\alpha,\beta} + \mathbf{t}(x^\beta_0) - R_{(x^\beta_0)} p_{1:1}^{\beta,\alpha} - \mathbf{t}(x^\beta_0) + N(0, \Sigma_{1:1}^{\alpha,\beta}), \alpha = 2, ..., M - 1, \end{cases}$$

(5.3)

where $\beta = \alpha + 1, ..., M$, $p_{1:1}^{\alpha,\beta} = [p_1^{\alpha,\beta}, ..., p_k^{\alpha,\beta}]$, $\mathbf{t}(x^\beta_0) = [(x^\alpha_1), ..., (x^\alpha_k)]_{3 \times k}$, and $N(0, \Sigma_{1:1}^{\alpha,\beta})$ is a zero-mean Gaussian random vector with covariance matrix $\Sigma_{1:1}^{\alpha,\beta} = diag(\Sigma_1^{\alpha,\beta}, ..., \Sigma_k^{\alpha,\beta})$, that models FLE in data sets, and $y_{1:k}$ is the column vector.

To estimate the state vector $x$, the following algorithm iterates through all pairs of corresponding points.

1) Predict the state vector $x$ and its covariance matrix $P_x$ from the state model in (5.2), as

$$\hat{x}_k^- = \hat{x}_{k-1}, \quad P_{\hat{x}_k^-} = P_{\hat{x}_{k-1}} + \Sigma_Q,$$

(5.4)
2) Append the $k$th pair of corresponding points from the overlapping data sets to the set of already selected pairs of corresponding points, and predict the corresponding points' positions in the reference frame using the computed state vector in Step 1, as
\[
\hat{y}_{1:k} = \begin{cases} 
R_{(\hat{x}_{\alpha}^\beta)} p_{1:k} + \hat{t}_{(\hat{x}_{\alpha}^\beta)}, \\
R_{(\hat{x}_{\beta}^\alpha)} p_{1:k} + \hat{t}_{(\hat{x}_{\beta}^\alpha)} - R_{(\hat{x}_{\beta}^\alpha)} p_{1:k} - \hat{t}_{(\hat{x}_{\beta}^\alpha)}, 
\end{cases}
\]
(5.5)

3) Calculate the distance error of the corresponding points in the reference frame to update the state vector and its covariance matrix as
\[
\hat{x}_k = \hat{x}_{-k} + K_k (y_{1:k} - \hat{y}_{1:k}), \quad P_{\hat{x}_k} = P_{\hat{x}_{-k}} - K_k E[y_{1:k}y_{1:k}^T] K_k^T, 
\]
(5.6)
where $K_k = E[x_{1:k}y_{1:k}^T]/E[y_{1:k}y_{1:k}^T]$ is the Kalman gain.

5.1.2 Experimental Validation

To verify the proposed algorithm, we performed an experiment on randomly generated data sets based on given correspondences. To do so, three overlapping data sets in the range of $\pm 100 \, mm$ were generated. Every data set consisted of 100 points, each with 50 overlapping points. Two rigid random transformations, in the range of $[\pm 50^\circ, \pm 50 \, mm]$, were applied to two of the data sets to bring them out of alignment. Three zero-mean Gaussian noise distributions with different variances were added to three overlapping areas among the data sets.

Then, the proposed algorithm was used to register the two data sets back to their original places. The procedure was repeated for 500 trials. We added different noise levels to data sets to verify the performance of the algorithm. We also compared the performance of the proposed technique with the one reported by Pennec [97].

Table 5.1 displays the variances considered for each overlapping area among the data sets, and the mean square surface registration error after global registration of the
Table 5.1: Surface registration error ($mm^2$) using the proposed and Pennec algorithms over 500 trials for three different experiments.

<table>
<thead>
<tr>
<th>Three Simulations</th>
<th>Surface registration error</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Overlap 1&amp;2</td>
<td>Overlap 1&amp;3</td>
<td>Overlap 2&amp;3</td>
</tr>
<tr>
<td>Variance of noise</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Pennec algorithm</td>
<td>0.96</td>
<td>0.98</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>UKF algorithm</td>
<td>0.97</td>
<td>0.99</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>Variance of noise</td>
<td>3</td>
<td>2</td>
<td>0.025</td>
<td></td>
</tr>
<tr>
<td>Pennec algorithm</td>
<td>2.91</td>
<td>1.93</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>UKF algorithm</td>
<td>2.97</td>
<td>1.95</td>
<td>0.027</td>
<td></td>
</tr>
<tr>
<td>Variance of noise</td>
<td>3</td>
<td>2</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>Pennec algorithm</td>
<td>2.9</td>
<td>1.92</td>
<td>0.12</td>
<td></td>
</tr>
<tr>
<td>UKF algorithm</td>
<td>2.96</td>
<td>1.95</td>
<td>0.004</td>
<td></td>
</tr>
</tbody>
</table>

As shown in Table 5.1, the proposed algorithm decreases the surface registration error in each overlapping area down to the variance of the noise added to that overlapping area. Figure 5.1 depicts the distance error histograms of the rotation and translation parameters over 500 trials where the variance of the noise added to each overlapping area was $1 \ mm^2$. As shown, the proposed algorithm converges to the true solution in all trials.

5.2 UKF Multi-body Surface-based Registration

In what follows, we extend the proposed UKF multi-body point-based registration algorithm to finely align multiple surfaces where their corresponding points are not
Figure 5.1: Error distribution of the estimated transformation parameters which simultaneously align the data sets 2 and 3 to the data set 1, using the proposed algorithm, over 500 trials.

given, but they are roughly aligned. The proposed algorithm assumes that the surfaces are nearly aligned in such a way that the closest point algorithm can return a reliable estimate of the true correspondences among the overlapping surfaces. Those correspondences are then used to globally register the surfaces using the following registration algorithm:

1) Choose three points from each surface set and find their corresponding points in the overlapping areas in the other surfaces;

2) Use the correspondences and the proposed UKF registration algorithm in Section 5.1, to register the surfaces;

3) Apply the estimated transformations to the surfaces; and

4) Append another point from each surface to the set of points already selected on that surface. Update the location of the corresponding points (using the closest
point metric) for all points considered in the optimization until this iteration, and proceed to Step 2.

This procedure iterates till it converges to a solution.

One application of the proposed multi-body surface-based registration algorithm is in computer-assisted orthopedic surgeries. Orthopedic surgeons are interested in registering multiple fractures of a bone back into their correct healing positions. Alignment of the bone fragments to construct a solid bone model is a complex problem that has not been highly investigated [51]. The main purpose of this problem is to take the surface representations of the fractured bones and attempt to determine how the broken surfaces fit together. The surfaces are usually generated from Computer Tomography (CT) images and may not be as high resolution as ideal surfaces, and therefore, they are perturbed by noise whose distribution may differ from one fracture area to another.

The problem of aligning bone fractures can be viewed as a multi-body surface registration problem where there are $M$ fragments that we would like to register by computing $M - 1$ transformations. There are a few works on bone and archeology fragment assembly in the literature [49, 95, 21, 137, 138], but, in the next subsection, we present a tentative solution and several numerical simulations showing that the proposed UKF multi-body surface-based registration algorithm can be employed to simultaneously and accurately register multiple bone fractures back to their correct healing positions.
5.2.1 Numerical Simulations

To analyze the performance of the proposed UKF multi-body surface-based registration algorithm, three independent simulations were performed. In the first simulation, the proposed algorithm is used to simultaneously register two femur bone fractures to the original femur bone (template). In the second simulation, three pieces of a fractured pelvic bone are registered to the original pelvic bone. Finally, in the last simulation, the proposed algorithm is used to register the two femur bone fractures back to their correct healing positions when there is no information about the original fractured bone and the fractures are not roughly aligned.

a) Registration of two femur bone fractures to the original femur bone:

In this experiment, three data sets collected from a fractured femur bone phantom are used. The first data set is generated by taking CT images of the femur bone, before it was fractured. CT images are captured using a LightSpeed Plus CT scanner. Then, Mesher software developed at our institution is used to semi-automatically segment the CT images, and to create a 3D surface model (bone template)—this contains 36,508 points—using a marching cubes algorithm. To produce the other two data sets, the femur bone phantom is fractured into two pieces. Another set of CT images is taken of each piece and a 3D surface model is then generated from that piece. The generated mesh of the bone and its fractures are shown in Figure 5.2.1.

Before fracturing the femur bone phantom, four fiducials were implanted on the bone surface. Those fiducials are used to register the bone fracture mesh data sets to the bone template. Two random transformations are then applied on the two fracture data sets to take them out of alignment. Next, the proposed UKF multi-body
surface-based registration algorithm is used to simultaneously register the transformed fracture data sets back to the bone template. To do so, 100 points are randomly chosen from each overlapping area among fractures and the bone template. Those points are then used to register the fracture data sets back to their original locations for two cases: where there are no outliers among the fracture data sets (100 points of each fracture are registered to 100 points of the other fracture or the bone template), and where there are outliers among data sets (100 points of each fracture are registered to all the points of the other fracture or bone template). After performing registration, the rest of the fracture points which are not used in the registration algorithm are employed to compute the surface registration error. This procedure is repeated for 250 trials for different ranges of random transformations. On average, each trial took 29 seconds to complete the registration.

Table 5.2 displays the convergence rate of the proposed algorithm in different
Table 5.2: Convergence rate of the proposed algorithm in globally registering multiple bone fractures to the bone template (fracture 1 to the template, fracture 2 to the template, and fracture 2 to 1) over 250 trails for different range of random transformations; when there is no outlier and when there are outliers in the data sets.

<table>
<thead>
<tr>
<th>Range of Transformation</th>
<th>Convergence rate in %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>with outlier</td>
</tr>
<tr>
<td>1-T</td>
<td>100</td>
</tr>
<tr>
<td>2-T</td>
<td>100</td>
</tr>
<tr>
<td>1-2</td>
<td>100</td>
</tr>
<tr>
<td>[±2.5 mm, ±2.5°]</td>
<td>99</td>
</tr>
<tr>
<td>[±5 mm, ±5°]</td>
<td>97</td>
</tr>
<tr>
<td>[±10 mm, ±10°]</td>
<td>82</td>
</tr>
<tr>
<td>[±15 mm, ±15°]</td>
<td>60</td>
</tr>
<tr>
<td>[±20 mm, ±20°]</td>
<td>48</td>
</tr>
</tbody>
</table>

simulation runs with varying ranges of transformations. It is assumed that the algorithm is able to converge for registration of each overlapping area when the mean surface registration error for that area is less than $\sqrt{2.5}$ mm. Figure 5.3 shows normalized histogram of surface registration error (in mm) for the three overlapping areas after performing registration over 250 trials for the range of transformation $[±2.5°, ±2.5$ mm]. Figure 5.4 depicts the registered fractured bone data sets to the bone template using the proposed algorithm for one of the simulation runs.

b) Registration of three pelvic bone fractures to the original pelvic bone:

In this simulation, we perform the same experiment as the previous one on a pelvic bone phantom. Using CT images, a 3D mesh is generated from the phantom, containing 83,562 points. Then, as shown in Figure 5.5, the bone is fractured into three pieces. Three surface meshes are generated by taking CT images of the fractured pieces. Before fracturing the phantom, some fiducials are mounted on the bone.
Those markers are used to register the fractures back to their original locations on the bone template. Three random transformations in the range of $[\pm 10^\circ, \pm 10 \text{ mm}]$ are applied to the fractures to bring them out of alignment. Then, the proposed UKF multi-body surface-based registration algorithm is used to simultaneously register 80 points, randomly selected from each fractures’ surface, to the bone template. This procedure is repeated for 250 trials. On average, each trial took 22 seconds to finish the registration.

Figure 5.6 displays the registered pelvic bone fractures to the pelvic bone template for a simulation run. Before and after performing registration, the mean surface registration error for all the overlapping areas over 250 trials are calculated to be 14.61 mm and 1.41 mm, respectively (with outliers); and 14.6 mm and 0.83 mm, respectively (with no outliers).
c) Registration of two femur bone fracture without template and initial alignments:

In the previous experiments, it is assumed that the original bone template is available and also, the corresponding points among surfaces are either known, or the surfaces are close enough to the bone template such that the closest points among them are reliable candidates for the corresponding points. Unfortunately, this method has limited clinical applicability since, in a real surgical scenario, the original bone template is not generally available. In addition, the corresponding points among fractures and the template are not known, furthermore, it may not be trivial to initially manually align the fractures in a surgical planning environment such that the registration algorithm succeeds. In this experiment, we improve our previous simulations by replacing the original bone template with a statistical anatomical
There exist many different approaches in the literature to generate a statistical atlas from many three-dimensional shape models [22, 105, 122]. In this experiment, we use principal component analysis (PCA) [20] to generate the statistical atlas model from femur bones of 12 human cadavers. Appendix A briefly explains how PCA is employed to generate the statistical atlas model from 12 femur bones. Figure 5.7 depicts the generated statistical atlas model (mean shape).

Furthermore, to roughly align the fractures to the mean shape (coarse registration), we use a local point descriptor to automatically find the potential corresponding points among the fractures and the mean shape. There exist many different point descriptors in the literature (such as spin images [52], point signatures [18], structural indexing [118], shape contexts [39] and point fingerprint [121]), but in this experiment, we have defined a new and simple point descriptor, as explained in Appendix B, to find the potential correspondences. This descriptor helps finding the potential correspondences among the fractures and the mean shape. Those correspondences are then used to perform a coarse registration among the fractures and the mean shape.

After alignment of the fractures to the mean shape, the proposed UKF multi-body surface-based registration algorithm is utilized to tune the alignment registration. Figure 5.8 shows the registered femur bone fractures to the mean shape after finding the matching points among the data sets and performing coarse and fine registration, respectively. The mean surface registration error between the two fractures, after performing registration, reduced to 1 mm which nearly matches the mean surface registration error between between fracture one and two in Simulation 1, as shown in
Figure 5.3. It means that both performed registrations in Simulations 1 and 3 have converged to the same point\(^1\).

![Figure 5.7: Generated mean shape from 12 femur bone cadaver data sets.](image)

![Figure 5.8: Registration of two fractures of a femur bone phantom to the mean shape.](image)

5.3 Summary and Discussion

The goal of this chapter was to present a new multi-body registration algorithm for registering multiple overlapping data sets and surfaces based on known and unknown correspondences.

In the case of known correspondences, a new multi-body registration algorithm based on UKF was presented. The proposed algorithm was used to simultaneously register randomly generated data sets, and its performance was compared with the Pennec algorithm [97]. As shown in Figure 5.1, the proposed algorithm accurately registered all the randomly generated data sets together such that the error of the estimated translations and rotations parameters are less than 1 mm and 1°, respectively.

\(^1\)For a detailed discussion on the performance of the proposed point descriptor, one can refer to [85].
Table 5.1 shows the comparison between the performance of the Pennec [97] and proposed algorithms in registering three overlapping data sets. As shown, the proposed algorithm was able to reduce the mean-square surface registration error among the overlapping areas to the level of the variance of the noise added to each data set. This means that, in the examined scenario, the proposed algorithm is more accurate than the Pennec technique.

The proposed multi-body point-based registration algorithm was then extended to simultaneously register multi surfaces where the corresponding points among them were not available. In the proposed UKF multi-body surface-based algorithm, it was assumed that the surfaces are roughly aligned such that closest point algorithm can be used to compute the corresponding points. Figures 5.4 and 5.6 illustrate how the proposed algorithm is able to accurately register multi-fracture bone surfaces of femur and pelvic bone phantoms back to their original positions, respectively.

A problem of the proposed algorithm is that it is sensitive to the initial alignment of the surfaces. If the surfaces are not closely aligned then the proposed registration algorithm can not accurately register the surfaces. Table 5.2 shows the sensitivity of the proposed registration algorithm to the initial alignments. As it is shown, the convergence rate of the proposed registration algorithm reduces as the initial alignment error among the data sets increases.

Finally, the proposed UKF multi-body surfaced-based registration algorithm was used to solve a clinical orthopedic problem where there are multi-bone fractures that should be registered back to their healing positions. The original bone of the fractures was not available and the fractures were not roughly and preliminarily aligned.

To register the fractures back to their correct position, first we generated a dummy
template for the fractured bones. We used PCA to produce a statistical mean shape of the original fractured bone. Then, the mean shape was used as a template to which the fractures should be registered.

Also, to roughly align the fractures to the mean shape, a local point descriptor was introduced that simply computes the potential matching points among the fractures and the mean shape. Finally, the proposed UKF multi-body surface-based registration algorithm was used to tune the registration. Figure 5.8 shows the registration of two femur bone fractures to a mean shape, shown in Figure 5.7, that is generated from 12 femur bones of human cadavers. As Figure 5.7 shows, the proposed algorithm is able to accurately register femur bone fractures which were not part of the bones used for generating the mean shape, by reducing the mean surface registration error between the fractured surfaces down to 1 mm.

To this end, we presented different registration algorithms to register medical data sets, and the performance of the proposed algorithms were compared with their peer algorithms in the literature. But, we did not discuss how accurate the performed registration is at the points which are not used in the registration process. In the next chapter, we present new solutions to estimate the accuracy of the performed registrations at any arbitrary points in the data sets.
Chapter 6

Measurement of the Registration Accuracy

Many researchers have successfully estimated the mean-squared value of TRE [36, 71, 114, 133]. Fitzpatrick and West [36], based on the Sibson’s formulations [113], mathematically derived an expression for the mean-squared value of TRE when FLE has an identical isotropic zero-mean Gaussian distribution; however, for computing the distribution of TRE, they used the Monte-Carlo simulation.

In this chapter, for the first time, we drive closed-form solutions that estimate the distribution of TRE and the mean-squared value of TRE when FLE has an arbitrary zero-mean Gaussian distribution. We also mathematically present two expressions that respectively indicate the loci of the points having the same mean-squared value of TRE, and the points at which the mean-squared value of TRE is minimum. In addition, we show that when FLE has an identical isotropic zero-mean Gaussian distribution, the variance of the transformation parameters (computed by the UKF registration algorithm) can be used in the proposed formulations to compute the
distribution of TRE, mean-squared value of TRE, the loci of the points having the same mean-squared value of TRE, and the points at which the mean-squared value of TRE is minimized.

By using ML algorithm, we generalize the proposed derivations to estimate the distribution of TRE and mean-squared value of TRE when FLE has an arbitrary distribution. We mathematically show that the proposed general solutions match the ones presented earlier in the literature for the special cases where FLE has an inhomogeneous and anisotropic or identical and anisotropic or identical and isotropic zero-mean Gaussian distribution, respectively.

The main achievements of this chapter are as follows:

1) Closed-form solutions that estimate the distribution of TRE, mean-squared value of TRE, the loci of the points having the same mean-squared value of TRE, and the points at which mean-squared value of TRE is minimized, are presented.

2) When FLE has an identical isotropic zero-mean Gaussian distribution, the variance of the transformation parameters calculated by the UKF registration algorithm, is used in the proposed closed-form solutions to estimate the TRE characteristics.

3) When FLE has an arbitrary distribution, the ML algorithm is used to calculate the variance of the transformation parameters, and estimate the TRE characteristics using the proposed closed-form solutions.

4) When FLE has an identical isotropic zero-mean Gaussian distribution, all the proposed algorithms in the literature estimating the mean-squared value of TRE, converge to the proposed ML solution.

In the remainder of this chapter, we first discuss the estimation of TRE characteristics in the presence of identical isotropic zero-mean Gaussian FLE and then we
extend our discussion the the cases where FLE has an arbitrary distribution.

6.1 Isotropic FLE Distribution

In this section, we first briefly explain the Fitzpatrick and West algorithm for estimation of distribution of TRE. Then, our proposed method for this purpose is presented. Finally, the performance of the proposed algorithm is verified and compared with the Fitzpatrick and West algorithm.

6.1.1 Fitzpatrick and West Algorithm

Assume there is one data set (moving data set) $U$, with size $N \times 3$ containing $N$ three-dimensional points. The centroid of $U$ is assumed to be at the origin of that data set. The fixed data set is generated by disturbing the moving data set with a perturbation matrix $F$ as follows:

$$Y = U + \epsilon F,$$  \hspace{1cm} (6.1)

where $F$, representing FLE, is assumed to be a matrix with the size of $N \times 3$, whose elements are independent zero-mean Gaussian random variables with a variance of $\sigma^2$. Thus, FLE is assumed to have an identical distribution at each fiducial point, and to be isotropic. $\epsilon$ is a positive dimensionless constant whose value will be taken to be small enough to control nonlinearities.

Since FLE is assumed to be isotropic, without a loss of generality, FLE in the data set $U$ can be assumed to be zero. This can be accomplished by simply replacing the variance of FLE in the data set $Y$ with the sum of the variances of FLE in the moving and fixed data sets [61], [82]. Moreover, because the target registration error is
independent of reorientation and FLE is isotropic, to estimate the TRE distribution, it is always possible to simply register the data set \( U \) to the data set \( Y \) as formulated in (6.1) [36].

To register the moving data set \( U \) to the fixed data set \( Y \), the registration algorithm proposed by Arun [2] is used to estimate the rotation matrix \( R \) and the translation vector \( t \). After estimating the registration parameters, the target registration error at the target point \( r = [r_x, r_y, r_z]^T \) has been expressed as

\[
e_r = \hat{R}r + \hat{t} - r,
\]

where \( \hat{R} \) and \( \hat{t} \) are the estimated rotation matrix and translation vectors, respectively. In [36], it has been shown that \( \hat{t} \) is linear in terms of \( \epsilon \) and can be computed as

\[
\hat{t} = \frac{1^T_N F}{N},
\]

where \( 1_N \) is a \( N \times 1 \) vector whose elements are all one. But \( \hat{R} \) has been shown to be nonlinear. Hence, to work with the linear terms of \( \epsilon \), the rotation matrix \( R \) has been expressed as a power series with respect to \( \epsilon \) as follows [36]:

\[
\hat{R} = I + \epsilon \hat{R}^1 + O(\epsilon^2).
\]

By substituting (6.4) into (6.2), and ignoring the higher order terms of \( \epsilon \), one can write

\[
e_r = \epsilon \hat{R}^1 r + \hat{t},
\]

where \( \hat{R}^1 \) can be easily computed from \( U \) and \( F \) as [36]

\[
\hat{R}^1_{ij} = \frac{\Lambda_{ii} Q_{ij} - \Lambda_{jj} Q_{ji}}{\Lambda^2_{ii} + \Lambda^2_{jj}},
\]

\[
Q = \Gamma^T (F - \frac{1_N^T 1_N}{N} F),
\]

where \( \hat{R}^1_{ij} \) and \( Q_{ij} \) are the \( ij \)th entries of the matrices \( \hat{R} \) and \( Q \), respectively. \( \Lambda_{ii} \) is
the $i$th entry of the diagonal matrix $\Lambda$ which contains the eigenvalues of $U$, and $\Gamma$ is an orthogonal matrix containing the eigenvectors of $U$. To determine $\Lambda$ and $\Gamma$, one should pick the orientation of the data set $U$ along the principal axes of $U$, and then compute SVD of $U$ [36] as

$$U = \Gamma \Lambda. \quad (6.8)$$

Having computed $\Gamma$ and $\Lambda$ from (6.8), $Q$ and $\hat{R}^1$ can be derived from (6.6) and (6.7), respectively. Since $\hat{R}^1$ is an antisymmetric matrix (it can be easily verified by looking at (6.6)), one can reformulate (6.5) as [36]

$$e_r = \hat{\Omega} \times r + \hat{t}, \quad (6.9)$$

where

$$\hat{\Omega} = [\hat{R}^1_{23}, -\hat{R}^1_{13}, \hat{R}^1_{12}]^T. \quad (6.10)$$

In [36], $\hat{t}$ and $\hat{\Omega}$ have been shown to be independent zero-mean Gaussian random vectors with covariance matrices $\Sigma_t = E[\hat{t}\hat{t}^T]$, and $\Sigma_{\Omega} = E[\hat{\Omega}\hat{\Omega}^T]$, respectively, as

$$\Sigma_t = \begin{bmatrix} \frac{\sigma^2}{N} & 0 & 0 \\ 0 & \frac{\sigma^2}{N} & 0 \\ 0 & 0 & \frac{\sigma^2}{N} \end{bmatrix}, \quad (6.11)$$

$$\Sigma_{\Omega} = \begin{bmatrix} \frac{\sigma^2}{\Lambda_{22}^2 + \Lambda_{33}^2} & 0 & 0 \\ 0 & \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{33}^2} & 0 \\ 0 & 0 & \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{22}^2} \end{bmatrix}. \quad (6.12)$$

e_r has been then resolved into three components $e_r$, $e_v$, and $e_w$ along three orthogonal vectors $\bar{r}$, $\bar{v}$ and $\bar{w}$ such that each of these three components is independent of each
other [36], as
\[
e_r = e_r \bar{r} + e_v \bar{v} + e_w \bar{w},
\]
(6.13)
\[
E[e_r e_v] = E[e_r e_w] = e[e_v e_w] = 0.
\]
(6.14)
\[
\bar{r} \text{ has been considered to be } \frac{r}{|r|}, \text{ where } |r| \text{ is the magnitude of vector } r. \ \bar{v} \text{ and } \bar{w} \text{ have been mathematically determined such that (6.14) has been satisfied. Using (6.13), it has been shown that [36]}
\]
\[
e_r = e_r \bar{r} = \hat{t} \bar{r},
\]
(6.15)
\[
e_v = e_v \bar{v} = (\hat{\Omega} \times r) \bar{v} + \hat{t} \bar{v},
\]
(6.16)
\[
e_w = e_w \bar{w} = -|r|(\bar{v} \hat{\Omega}) + \hat{t}(\bar{r} \times \bar{v}).
\]
(6.17)

Then, the mean-squared value of TRE is estimated by using (6.13) as
\[
E[TRE_r^2] = E[e_r^T e_r] = E[e_r^2] + E[e_v^2] + E[e_w^2],
\]
(6.18)
where \(E[e_r^2]\), \(E[e_v^2]\), and \(E[e_w^2]\) are determined using (6.15), (6.16), and (6.17), respectively, as
\[
E[e_r^2] = \frac{\sigma^2}{N},
\]
(6.19)
\[
E[e_v^2] = |r|^2 E[(\hat{\Omega} \bar{w})^2] + \frac{\sigma^2}{N},
\]
(6.20)
\[
E[e_w^2] = |r|^2 E[(\hat{\Omega} \bar{v})^2] + \frac{\sigma^2}{N}.
\]
(6.21)

By substituting (6.19), (6.20), and (6.21) into (6.18), (6.18) is reformulated as follows:
\[
E[TRE_r^2] = \frac{3\sigma^2}{N} + |r|^2 (E[(\hat{\Omega} \bar{v})^2] + E[(\hat{\Omega} \bar{w})^2]),
\]
\[
= \frac{3\sigma^2}{N} + |r|^2 (E[|\hat{\Omega}|^2] - E[(\hat{\Omega} \bar{r})^2]).
\]
(6.22)
By simplifying (6.22), one has \[36\]
\[
E[TRE_r^2] = \frac{3\sigma^2}{N} + \frac{(r_y^2 + r_z^2)}{\Lambda_{22}^2 + \Lambda_{33}^2} \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{33}^2} + \frac{(r_x^2 + r_z^2)}{\Lambda_{11}^2 + \Lambda_{22}^2} \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{22}^2}
\]
\[= \frac{3\sigma^2}{N} + \frac{r_x^2}{\Lambda_{11}^2 + \Lambda_{33}^2} \left( \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{22}^2} \right) + \frac{r_y^2}{\Lambda_{22}^2 + \Lambda_{33}^2} \left( \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{22}^2} \right) + \frac{r_z^2}{\Lambda_{22}^2 + \Lambda_{33}^2} \left( \frac{\sigma^2}{\Lambda_{11}^2 + \Lambda_{22}^2} \right),
\]
(6.23)

Finally, after computing the mean-squared value of TRE, Fitzpatrick and West \[36\] used the Monte-Carlo simulation to derive the distribution of TRE, i.e., the distribution of |e(r)|.

As shown in (6.14), \(e_r, e_v,\) and \(e_w\) are uncorrelated and since they have Gaussian distributions, they are independent. Therefore, the distribution of \(TRE_r, |e(r)|,\) can be computed by taking the square root of the summation of the three squared Gaussian random variables, followed by the Monte-Carlo simulation \[36\]
\[TRE_r \sim \sqrt{e_r^2 + e_v^2 + e_w^2}.\]
(6.24)

6.1.2 Proposed Algorithm based on UKF

Let us assume that the registration is performed between two data sets, and rotation matrix \(R\) and translation vector \(t\) are accurately estimated by the proposed point-based UKF registration algorithm as \(\hat{R} \triangleq R\hat{x}_\theta\) and \(\hat{t} \triangleq \hat{x}_t\), respectively.

Now, let \(r\) be the point, other than the ones used in the registration algorithm, at which we would like to compute TRE. The registration error at the target point \(r\) can be determined as

\[e_r = \hat{R}r + \hat{t} - Rr - t = (\hat{R} - R)r + (\hat{t} - t),\]
(6.25)
\[= \Delta Rr + \Delta t,\]
where

\[
\Delta R = \hat{R} - R = R_{x_{\theta}} - R_{\hat{x}_{\theta}}, \quad (6.26)
\]

\[
\Delta t = \hat{t} - t = \hat{x}_{t} - x_{t}. \quad (6.27)
\]

By this assumption that the registration algorithm has converged to the true values of transformation parameters (\(\Delta R\) and \(\Delta t\) are assumed to be unbiased), based on (4.5), \(\Delta t\) is a zero-mean Gaussian random vector with the covariance matrix \(\Sigma_t\) that can be easily derived from (4.23) as follows:

\[
P_\hat{x} = E[(x - \hat{x})(x - \hat{x})^T] = \begin{pmatrix}
\Sigma_{x_t} & \Sigma_{x_t x_{\theta}} \\
\Sigma_{x_t x_{\theta}} & \Sigma_{x_{\theta}}
\end{pmatrix}, \quad (6.28)
\]

where

\[
\Sigma_{x_t} = E[(x_t - \hat{x}_t)(x_t - \hat{x}_t)^T], \quad (6.29)
\]

\[
\Sigma_{x_t x_{\theta}} = E[(x_t - \hat{x}_t)(x_{\theta} - \hat{x}_{\theta})^T], \quad (6.30)
\]

\[
\Sigma_{x_{\theta}} = E[(x_{\theta} - \hat{x}_{\theta})(x_{\theta} - \hat{x}_{\theta})^T]. \quad (6.31)
\]

\(\Delta R_r\), in (6.26), is a random vector with an unknown distribution. However, its mean is zero \(E[\Delta R_r] = 0\) and its covariance matrix \(\Sigma_{R_r} = E[\Delta R_r(\Delta R_r)^T]\) can be estimated using the UT explained in Chapter 3, because the rotation matrix \(R\) is a nonlinear function of the Gaussian random vector \(\mathbf{x}_\theta\) with mean of \(\hat{\mathbf{x}}_\theta\) and covariance matrix of \(\Sigma_{\mathbf{x}_\theta}\). \(\hat{\mathbf{x}}_\theta\) and \(\Sigma_{\mathbf{x}_\theta}\) can be determined from (5.6) and (4.22) or (6.28), respectively.

Since \(\Delta t\) is assumed to be a zero-mean Gaussian random vector and \(\Delta R_r\) is a zero-mean random vector with an unknown distribution that might have correlation with \(\Delta t\), one can approximately assume that the error vector \(e_r\) has a zero-mean
Gaussian distribution with the covariance matrix of $E[e_re_r^T]$ as follows:

$$E[e_re_r^T] = E[(\Delta Rr + \Delta t)(\Delta Rr + \Delta t)^T],$$

$$= E[\Delta Rr(\Delta Rr)^T] + E[(\Delta Rr)\Delta t^T] + E[\Delta t(\Delta Rr)^T] + E[\Delta t\Delta t^T].$$ (6.32)

If the rotation and translation parameters were independent, one could assume that $E[(\Delta Rr)\Delta t^T]$ would be zero. However, here it is assumed that the translation and rotation parameters might be dependent and $\Sigma_{(Rr)t} = E[(\Delta Rr)\Delta t^T]$ may not be zero. Therefore, (6.32) can be reformulated as

$$E[e_re_r^T] = \Sigma_{Rr} + \Sigma_{(Rr)t} + \Sigma_{(Rr)t}^T + \Sigma_{xt}. \quad (6.33)$$

$\Sigma_{(Rr)t}$ is called the covariance matrix between $(\Delta Rr)$ and $\Delta t$. It can be computed by using the UT and propagating the estimated state vector $\hat{x}$ and its covariance matrix $P_{\hat{x}}$ through the nonlinearity $(\Delta Rr)\Delta t^T$.

Therefore, by calculating $\Sigma_{xt}$, $\Sigma_{(Rr)t}$ and $\Sigma_{Rr}$, the covariance matrix of the zero-mean Gaussian random vector $e_r$ can be derived from (6.33). Now, by defining $e_r$ as $[e_x, e_y, e_z]^T$, one can write $TRE_r$, the target registration error at point $r$, in terms of the error vector $e_r$ as follows:

$$TRE_r = \sqrt{(e_x^2 + e_y^2 + e_z^2)}, \quad (6.34)$$

where $e_x$, $e_y$, and $e_z$ are zero-mean Gaussian random variables.

Since $TRE_r$ is distance error, and distance is invariant to rotation, then the error vector $e_r$ can be arbitrarily rotated in a space without varying the distribution of $TRE_r$. Hence, one may rotate the error vector $e_r$ such that its covariance matrix, $E[e_re_r^T]$ becomes diagonal, and therefore, $e_x$, $e_y$, and $e_z$ become independent Gaussian random variables. Let us assume that $\Sigma_{e_r}$ is the diagonalized covariance matrix of
$E[e_\mathbf{e}^T]$ as follows:

$$
\Sigma_{e_\mathbf{e}} = \begin{pmatrix}
\sigma_{n_x}^2 & 0 & 0 \\
0 & \sigma_{n_y}^2 & 0 \\
0 & 0 & \sigma_{n_z}^2
\end{pmatrix}.
$$

(6.35)

where $\sigma_{n_x}^2$, $\sigma_{n_y}^2$, and $\sigma_{n_z}^2$ are the variance of the first, second and third components of the rotated error vector $\mathbf{e}$, respectively.

Since the distribution of $\text{TRE}_r$ can be directly derived from the distribution of $\text{TRE}^2_r$, and also, due to the simplicity, we first determine the distribution of $\text{TRE}^2_r$. $\text{TRE}^2_r$ is the summation of three squared independent zero-mean Gaussian random variables with variances of $\sigma_{n_x}^2$, $\sigma_{n_y}^2$, and $\sigma_{n_z}^2$, respectively, along three orthogonal axes. It means that the probability distribution function (pdf) of $\text{TRE}^2_r$ is the convolution of three squared Gaussian random variables as follows:

$$
\text{pdf}_{\text{TRE}^2_r}(x) = \text{pdf}_{(n_x^2)}(x) \ast \text{pdf}_{(n_y^2)}(x) \ast \text{pdf}_{(n_z^2)}(x)
$$

$$
\text{pdf}_{(n_x^2+n_y^2)}(x) \ast \text{pdf}_{(n_z^2)}(x),
$$

(6.36)

where ($\ast$) is the convolution operator, and $\text{pdf}_{(n_x^2+n_y^2)}(x)$ is defined to be $\text{pdf}_{(n_x^2)}(x) \ast \text{pdf}_{(n_y^2)}(x)$.

To do the convolutions, first the pdf of a squared Gaussian random variable should be calculated. It is easy to show that if $n_x$ has a Gaussian distribution with mean of zero and variance of $\sigma_{n_x}^2$, then the distribution of $n_x^2$ is

$$
\text{pdf}_{n_x^2}(x) = \frac{1}{\sqrt{2\pi x\sigma_{n_x}^2}} \exp\left(-\frac{x}{2\sigma_{n_x}^2}\right) \ x \geq 0.
$$

(6.37)

Consequently, the distribution of $\text{pdf}_{(n_x^2+n_y^2)}(x)$ can be computed as follows:

$$
\text{pdf}_{(n_x^2+n_y^2)}(x) = \text{pdf}_{n_x^2}(x) \ast \text{pdf}_{n_y^2}(x) =
$$

$$
\frac{1}{\sqrt{2\pi x\sigma_{n_x}^2}} \exp\left(-\frac{x}{2\sigma_{n_x}^2}\right) \ast \frac{1}{\sqrt{2\pi x\sigma_{n_y}^2}} \exp\left(-\frac{x}{2\sigma_{n_y}^2}\right) \ x \geq 0,
$$

(6.38)
which can be more simplified as

\[
pdf_{n_x^2 + n_y^2}(x) = \frac{1}{2\sigma_n x \sigma_n y} \exp\left[-\frac{x}{4} \left(\frac{1}{\sigma_n^2 x} + \frac{1}{\sigma_n^2 y}\right)\right] I_o\left(\frac{x}{4} \left(\frac{1}{\sigma_n^2 x} - \frac{1}{\sigma_n^2 y}\right)\right) \quad x \geq 0,
\]

where \( I_o(x) \) is the modified Bessel function

\[
I_o(x) = \frac{1}{\pi} \int_0^\pi \exp(x \phi)d\phi.
\]

Finally, the distribution of \( TRE_r^2 \) can be obtained from (6.36) as follows:

\[
pdf_{TRE_r^2}(x) = \frac{1}{2\sqrt{2\pi}\sigma_n x \sigma_n y \sigma_n z} \exp\left(-\frac{x^2}{2\sigma_n^2 z}\right) \times \int_0^\infty \frac{\exp\left(\frac{\lambda}{4} \left[\frac{1}{\sigma_n^2 x} + \frac{1}{\sigma_n^2 y} - \frac{2}{\sigma_n^2 z}\right]\right)}{\sqrt{(x^2 - \lambda)}} I_o\left(\frac{\lambda}{4} \left[\frac{1}{\sigma_n^2 x} - \frac{1}{\sigma_n^2 y}\right]\right)d(\lambda).
\]

At last, the probability distribution function of \( TRE_r = \sqrt{n_x^2 + n_y^2 + n_z^2} \) can be determined as\(^1\)

\[
pdf_{TRE_r}(x) = (2x) \times pdf_{TRE_r^2}(x^2) = \frac{x}{\sqrt{2\pi}\sigma_n x \sigma_n y \sigma_n z} \exp\left(-\frac{x^2}{2\sigma_n^2 z}\right) \times \int_0^\infty \frac{\exp\left(\frac{\lambda}{4} \left[\frac{1}{\sigma_n^2 x} + \frac{1}{\sigma_n^2 y} - \frac{2}{\sigma_n^2 z}\right]\right)}{\sqrt{(x^2 - \lambda)}} I_o\left(\frac{\lambda}{4} \left[\frac{1}{\sigma_n^2 x} - \frac{1}{\sigma_n^2 y}\right]\right)d(\lambda).
\]

Now that the distribution of TRE at an arbitrary target location is estimated, by following the same procedure we can derive a closed-form formula for the mean-squared value of TRE, the point at which the mean-squared value of TRE is minimized and the loci of the points having the same mean-squared values of TRE, respectively.

\( a) \) **Estimating the mean-squared value of TRE:**

By using (6.25), the mean-squared value of \( TRE_r \) can be formulated as

\[
E[TRE_r^2] = E[\mathbf{e}_r^T \mathbf{e}_r] = E[(\Delta \mathbf{R} \mathbf{r} + \Delta \mathbf{t})^T (\Delta \mathbf{R} \mathbf{r} + \Delta \mathbf{t})],
\]

\[
= \mathbf{r}^T \Sigma_{\mathbf{R} \mathbf{r}} \mathbf{r} + 2\mathbf{v}^T \mathbf{r} + E[\Delta \mathbf{t}^T \Delta \mathbf{t}], \tag{6.42}
\]

\(^1\)It should be mentioned that (6.41) is also valid for estimation of the distribution of TRE when FLE has an arbitrary Gaussian distribution.
where \( \mathbf{v} \) is defined to be \( E[\Delta \mathbf{t}^T \Delta \mathbf{R}]^T \), and \( \Sigma_R = E[\Delta \mathbf{R}^T \Delta \mathbf{R}] \) is the covariance matrix of \( \Delta \mathbf{R} \) that is a symmetric positive definite matrix. Also, \( E[\Delta \mathbf{t}^T \Delta \mathbf{t}] \) can be written as \( E[\Delta \mathbf{t}^T \Delta \mathbf{t}] = E[\Delta t_x^2] + E[\Delta t_y^2] + E[\Delta t_z^2] \), where \( E[\Delta t_x^2] \), \( E[\Delta t_y^2] \) and \( E[\Delta t_z^2] \) are the main diagonal entries of the covariance matrix \( \Sigma_x \) as
\[
E[\Delta t_x^2] = \Sigma_{x_1}^{11}, \quad E[\Delta t_y^2] = \Sigma_{x_2}^{22}, \quad E[\Delta t_z^2] = \Sigma_{x_3}^{33},
\]
where \( \Sigma_{x_i}^{ij} \) is the \( ij \)-th entry of the covariance matrix \( \Sigma_x \).

Also, since \( \mathbf{R} \) is a nonlinear function of the Gaussian random vector \( \mathbf{x}_\theta \) with mean of \( \hat{\mathbf{x}}_\theta \) and covariance matrix of \( \Sigma_x \theta \), \( \Sigma_R \) can be estimated using the UT. Vector \( \mathbf{v} \) can also be computed by using the UT and propagating \( \hat{\mathbf{x}}_r \) and \( \mathbf{P}_x \) through the nonlinearity \( (\Delta \mathbf{t}^T \Delta \mathbf{R}) \).

Therefore, by estimating the variance of translation components, \( \Sigma_R \) and \( \mathbf{v} \), \( E[TRE_r^2] \) can be obtained from (6.42).

Here, two interesting questions arise. First, at which location \( \mathbf{r} \) is the mean-squared value of \( TRE_r \) minimized and what is the value of mean-squared \( TRE_r \) at that location? Second, what are the loci of the points that have the same mean-squared value of \( TRE \)? These questions are answered in what follows.

b) Minimum mean-squared value of \( TRE \):

From (6.42), we see that the mean-squared value of \( TRE_r \) is the summation of five components. The last three components are independent of the target location \( \mathbf{r} \), and are constant. The first two terms are quadratic terms with respect to \( \mathbf{r} \). To find the minimum mean-squared value of \( TRE_r \) and the point at which this minimum is achieved, the derivative of (6.42) with respect to \( \mathbf{r} \) is computed as
\[
\frac{\partial E[TRE_r^2]}{\partial \mathbf{r}} = 2\Sigma_R \mathbf{r} + 2\mathbf{v}.
\]
To find the point at which the mean-squared value of TRE is minimized, one should set (6.44) to zero and solve for \( r \) as follows:

\[
E_r^2 = -\Sigma^{-1}_R v = -E[\Delta R^T \Delta t] / E[\Delta R^T \Delta R]. \tag{6.45}
\]

If there is no correlation between \( \Delta R \) and \( \Delta t \) (the assumption made by Fitzpatrick and West [36]), then \( r_{\text{min}} \) turns to be zero. This result can also be verified by looking at (6.23) which is minimized when the target is at the origin, \( r = 0 \).

The minimum mean-squared value of \( TRE_r \) can be easily found by substituting (6.45) in (6.42) as

\[
\min \{E[TRE_r^2]\} = \mathbb{E}[\Delta t^2] + E[\Delta t_y^2] + E[\Delta t_z^2] - v^T \Sigma^{-1}_R v. \tag{6.46}
\]

c) Locus of the points with the same mean-squared TRE:

To find the locus of the points with the same mean-squared value of TRE, assume that \( v \) and the estimated symmetric covariance matrix \( \Sigma_R \) have the following formats

\[
\Sigma_R = \begin{bmatrix}
\Sigma_{11}^R & \Sigma_{12}^R & \Sigma_{13}^R \\
\Sigma_{12}^R & \Sigma_{22}^R & \Sigma_{23}^R \\
\Sigma_{13}^R & \Sigma_{23}^R & \Sigma_{33}^R
\end{bmatrix}, \quad v = [v_1, v_2, v_3]^T. \tag{6.47}
\]

By substituting (6.47) into (6.42), and also, by representing the target point \( r \) as the vector \([r_x, r_y, r_z]^T\), (6.42) can be simplified as

\[
E[TRE_r^2] = \Sigma_{11}^R r_x^2 + \Sigma_{22}^R r_y^2 + \Sigma_{33}^R r_z^2 + 2\Sigma_{12}^R r_x r_y
\]
\[
+ 2\Sigma_{13}^R r_x r_z + 2\Sigma_{23}^R r_y r_z + 2v_1 r_x + 2v_2 r_y + 2v_3 r_z + E[\Delta t_x^2] + E[\Delta t_y^2] + E[\Delta t_z^2]. \tag{6.48}
\]

Equation (6.48) demonstrates that the loci of the points, which have the same \( E[TRE_r^2] \), are ellipsoids.
6.1.3 Numerical Simulations

In order to test our approximations and verify our derivations, specifically (6.41), (6.42), and (6.48), several simulations on random and real data sets are performed. First, we run the simulations on the data sets which are randomly generated from uniform distributions. Then, the data sets generated from a scaphoid and a shoulder bone phantom, and a pelvic cadaver bone are employed to examine the derived equations.

a) Simulation results on randomly generated isotropic data sets:

In the first simulation, we chose $N$, the number of points in data sets, to be 30. The moving data set is generated by drawing $N$ fiducial points uniformly within a cube with the sides of $\pm 100\text{mm}$. Also, one target location ($r$) is selected randomly from a cube of sides $\pm 200\text{mm}$. In order to generate the fixed data set, we perturbed independently $x$, $y$, and $z$ components of each point in the moving data set by a zero-mean Gaussian random variable which models the FLE along each orthogonal axis. The distribution of FLE along each axis is considered to be $\mathcal{N}(0, 6\text{ mm}^2)$. In this way, we produced the same model as that used by Fitzpatrick and West [36].

Then, the moving data set is registered to the fixed one, and the target registration error (TRE) is measured using (6.2) at the target point. The distribution of TRE is numerically estimated by repetitions of perturbation and registration steps for 100,000 times. Also, just for one trial, the two data sets (moving and fixed data sets) are registered using the UKF registration algorithm to compute the variance of the estimated registration parameters. Equation (6.41) is then used to estimate the distribution of TRE at $r$. Figures 6.1 and 6.2 show the estimated probability and cumulative
distribution functions of TRE at the target point \( r = [140.2539, -117.0949, 95.4582]^T \) using the proposed algorithm, Fitzpatrick and West algorithm, and the one obtained by numerical simulations, respectively. As shown, both algorithms can accurately estimate the distribution of TRE at the target point \( r \).

![Probability Density](image)

Figure 6.1: Probability densities of the estimated TRE using numerical simulations proposed, and Fitzpatrick and West algorithms when the data sets have the same moments along each orthogonal axis.

Next, we compared the estimated TRE distributions with the simulated ones by computing the distance between the distributions. The distance between two distributions, \( f_1 \) and \( f_2 \) can be defined as [63]

\[
d(f_1, f_2) = \frac{1}{2} \left( \int_{-\infty}^{\infty} \left[ f_1(\lambda) \log \frac{2f_1(\lambda)}{f_1(\lambda) + f_2(\lambda)} + f_2(\lambda) \log \frac{2f_2(\lambda)}{f_1(\lambda) + f_2(\lambda)} \right] d(\lambda) \right). \tag{6.49}
\]

Using (6.49), the distance between the estimated TRE distributions and the simulated one is calculated and displayed in Table 6.1. Additionally, by using (6.42) and (6.23), the mean-squared value of TRE of the estimated distributions are calculated and listed in Table 6.2. As shown in Tables 6.1 and 6.2, the estimated distribution of
Figure 6.2: Cumulative distributions of the estimated TRE using numerical simulations proposed, and Fitzpatrick and West algorithms when the data sets have the same moments along each orthogonal axis. The cumulative distribution generated by numerical simulations exactly fits the ones generated by the other methods.

TRE and estimated mean-squared value of TRE using both methods are very similar to those generated by numerical simulations.

We repeated this simulation for 250 trials. In each trial, the data sets and target location are randomly generated as explained earlier. Also, the mean-squared value of TRE at the chosen target is calculated using (6.42), (6.23), and the Monte-Carlo simulation. Table 6.3 shows the mean and variance of the estimation error of the mean-squared value of TRE over 250 trials for different variances of FLE perturbing the data sets. As shown, both Fitzpatrick and West, and the proposed algorithms’ results are identical in the precision shown.
Table 6.1: Distance between two distributions: 1- the distribution of TRE estimated by the Fitzpatrick and West method, and the one generated by numerical simulations; 2- The distribution of TRE generated by the proposed algorithm, and the one generated by numerical simulations.

<table>
<thead>
<tr>
<th>Experiments I&amp;II</th>
<th>Distance between distributions</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric data set</td>
<td>Fitzpatrick and West</td>
<td>0.04%</td>
<td>0.04%</td>
</tr>
<tr>
<td></td>
<td>Proposed</td>
<td>0.04%</td>
<td>0.04%</td>
</tr>
<tr>
<td>Asymmetric data set</td>
<td>0.8%</td>
<td>0.8%</td>
<td></td>
</tr>
</tbody>
</table>

Table 6.2: Computed mean-squared value of TRE using Fitzpatrick and West algorithm, proposed algorithm, and numerical simulations.

<table>
<thead>
<tr>
<th>Experiments I&amp;II</th>
<th>E[TRE$^2$] mm$^2$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric data set</td>
<td>3.33</td>
<td>3.38</td>
<td>3.41</td>
</tr>
<tr>
<td>Asymmetric data set</td>
<td>25.15</td>
<td>25.18</td>
<td>25.24</td>
</tr>
</tbody>
</table>
b) Simulation results on randomly generated anisotropic data sets:

In the second simulation, we do not generate the data sets uniformly in each direction. Instead, they have different moments along $x$, $y$, and $z$ directions. Basically, the three components of the points in the moving data set are generated randomly from uniform distributions with the sides of $\pm 5$ mm, $\pm 15$ mm, and $\pm 30$ mm, respectively. Then, the same simulation as the previous one is performed on the newly generated data sets. Figures 6.3 and 6.4 show the estimated TRE probability and cumulative distribution functions using the proposed algorithm, Fitzpatrick and West algorithm, and the one obtained by numerical simulations at the target point $\mathbf{r} = [26.29, 86.62, 4.52]^T$, respectively.

![Figure 6.3: Probability densities of the estimated TRE using numerical simulations, proposed algorithm, and Fitzpatrick and West algorithm when data sets have different moments along each orthogonal axis.](image)

As before, the distance between the estimated TRE distributions and the one generated by numerical simulations is shown in Table 6.1. Table 6.2 displays the mean-squared value of the TRE calculated by the Fitzpatrick and West algorithm,
Figure 6.4: Cumulative distributions of the estimated TRE using the proposed, and Fitzpatrick and West algorithms when data sets have different moments along each orthogonal axis. The cumulative distribution function generated by numerical simulations exactly fits the one generated by the proposed algorithm.

As illustrated in Tables 6.1 and 6.2, for this case where the datasets are not uniformly distributed along $x$, $y$, and $z$ axes, both the proposed method and the Fitzpatrick and West algorithm accurately estimate the TRE distribution at the target point $r$.

Again, this simulation is repeated for 250 trials. In each trial, the data sets and target are generated randomly and the mean-squared value of TRE at the chosen target is calculated. Table 6.4 displays the mean and variance of the estimation error of the mean-squared value of TRE over 250 trials for different variances of FLE perturbing the data sets. As shown, the Fitzpatrick and West results match those of the proposed algorithm. As can be seen, both algorithms’ performances are almost the same even for the case where the variance of the FLE is large.
Table 6.3: Mean (\(mm^2\)) and variance (\(mm^4\)) of error in estimating mean-squared value of TRE using the Fitzpatrick and West, and proposed algorithms over 250 trials when the registering data sets are symmetric.

<table>
<thead>
<tr>
<th>Covariance of FLE ((mm^2))</th>
<th>Fitzpatrick and West</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance</td>
</tr>
<tr>
<td>1 (\times \mathbf{I}_{3\times 3})</td>
<td>0.005</td>
<td>0.00004</td>
</tr>
<tr>
<td>10 (\times \mathbf{I}_{3\times 3})</td>
<td>0.0086</td>
<td>0.0001</td>
</tr>
<tr>
<td>20 (\times \mathbf{I}_{3\times 3})</td>
<td>0.071</td>
<td>0.0078</td>
</tr>
<tr>
<td>40 (\times \mathbf{I}_{3\times 3})</td>
<td>0.25</td>
<td>0.11</td>
</tr>
<tr>
<td>80 (\times \mathbf{I}_{3\times 3})</td>
<td>0.6</td>
<td>0.58</td>
</tr>
<tr>
<td>120 (\times \mathbf{I}_{3\times 3})</td>
<td>0.87</td>
<td>1.27</td>
</tr>
<tr>
<td>180 (\times \mathbf{I}_{3\times 3})</td>
<td>1.22</td>
<td>2.35</td>
</tr>
</tbody>
</table>

From the performed simulations, one can conclude that both algorithms are accurate in estimating the TRE distribution and its mean-squared value, with this difference that the proposed algorithm does not need the Monte-Carlo simulation to compute the distribution of TRE.

c) Simulation results on loci of the points having the same registration error:

In the third simulation, to verify (6.48), the loci of the points having the same mean-squared value of TRE are simulated. To do so, we generate the moving data set randomly from a cube with sides of \(\pm 100\ mm\). Also, a target position, \(\mathbf{p}_1 = [59.9569, -45.2688, -29.6124]^T\) is chosen randomly within a cube of sides \(\pm 200\ mm\). Then, a zero-mean Gaussian random noise, which models the FLE with the distribution of \(\mathcal{N}(0, 6\ mm^2\mathbf{I}_{3\times 3})\) is added to each point in the moving data set to generate the
Table 6.4: Mean ($mm^2$) and variance ($mm^4$) of error in estimating mean-squared value of TRE using the Fitzpatrick and West, and proposed algorithms over 250 trials when the registering data sets are asymmetric.

<table>
<thead>
<tr>
<th>Covariance of FLE ($mm^2$)</th>
<th>Fitzpatrick and West</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance</td>
</tr>
<tr>
<td>$1 \times I_{3 \times 3}$</td>
<td>0.001</td>
<td>0.000002</td>
</tr>
<tr>
<td>$10 \times I_{3 \times 3}$</td>
<td>0.7</td>
<td>4.6</td>
</tr>
<tr>
<td>$15 \times I_{3 \times 3}$</td>
<td>10.7</td>
<td>264</td>
</tr>
<tr>
<td>$25 \times I_{3 \times 3}$</td>
<td>90</td>
<td>12244</td>
</tr>
</tbody>
</table>

fixed data set. By using (6.48), the locus of the points having the same $z$ component as $p_1$ and the same mean-squared value of TRE as $p_1$ is calculated.

Figure 6.5 shows the loci of the points with the same mean-squared value of TRE. To verify the estimated loci of the points, we randomly chose another point, $p_2 = [0, 71.6955, -29.6124]^T$, from the obtained loci to estimate the distribution and mean-squared value of TRE at $p_2$ using numerical simulations and the proposed algorithm. Figure 6.6 shows the distributions of TRE at $p_1$ and $p_2$, respectively. Interestingly, as shown in Figure 6.6, not only the mean-squared value of TRE, but also the distribution of TRE are the same at $p_1$ and $p_2$.

d) Simulation results on pelvic cadaver bone:

In this simulation, data sets are generated by taking CT images of a pelvic cadaver bone. The resulting CT images are then segmented manually by using the commercial software AMIRA™ (TGS, Berlin, Germany) to produce a three-dimensional mesh
CHAPTER 6. MEASUREMENT OF THE REGISTRATION ACCURACY

Figure 6.5: Loci of the points with the same mean-squared value of TRE.

Figure 6.6: Probability densities of the target registration error at points $p_1$ and $p_2$ that are chosen from the same locus in Figure 6.5.
model containing 2,500 points. The fixed data set $Y$ is produced by randomly selecting 30 points from the pelvic surface, as shown in Figure 6.7. The moving data set $U$ is then generated by applying a random transformation to the chosen 30 points from the 3D mesh data set and adding zero-mean Gaussian noise modelling the FLE with covariance matrix $3 \, mm^2 I_{3 \times 3}$. The random transformation is generated by uniformly drawing each translation and rotation parameters within the cubes with sides of $\pm 80 \, mm$ and $\pm 80^\circ$, respectively.

![Figure 6.7: 3D surface model of the pelvic cadaver bone and the 30 selected points from the model, shown in black spheres, to do the registration.](image)

The UKF point-based registration algorithm is then used to register the moving data set to the fixed data set. After registration, the distance error of the rest of the 3D mesh points are measured based on the estimated registration parameters. This procedure is repeated 1,000 times, while keeping the chosen 30 points (the points used in the registration) from the 3D surface and the selected random transformation fixed. After 50,000 registration trials, the mean distance and mean-squared distance
error of each point at the 3D mesh are computed. Then, the proposed algorithm is employed to estimate the mean-squared value of TRE at each point on the 3D mesh.

Figure 6.8 shows the estimated mean-squared value of TRE, after registration, for each point on the 3D surface using the proposed algorithm. The estimated results are compared with the ones obtained by numerical simulations in 50,000 trails. Table 6.5 shows mean and variance of the estimation error of the mean-squared value of TRE and mean TRE at every point of the 3D mesh. As shown, the proposed algorithm is able to accurately estimate the mean and mean-squared value of TRE at any desirable point on the 3D mesh.

Figure 6.8: Estimated mean-squared value of TRE (in $mm^2$) at every point on the 3D surface model, generated from the pelvic cadaver bone, after registration.
e) Simulation results on scaphoid bone phantom:

In this simulation, data sets are generated by taking CT images of a scaphoid bone phantom. CT images are captured using a LightSpeed Plus CT scanner™ (GE Medical Systems, Waukesha WI). Mesher software developed at Queen’s University is used to semi-automatically segment the scaphoid bone phantom from the CT images, and to create a 3D surface model-containing 17,029 points-using a marching cubes algorithm. The fixed data set $Y$ is produced by randomly selecting 150 points from the 3D surface, as shown in Figure 6.9. The moving data set $U$ is then generated by applying a random transformation to the chosen 150 points from the 3D mesh and adding zero-mean Gaussian noise modelling the FLE with covariance matrix $0.1 \, mm^2 I_{3 \times 3}$. 1,000 random transformations, as before, are generated to take the moving data set out of alignment from the fixed data set. Then, the data sets are

![Figure 6.9: 3D surface model of the scaphoid bone phantom and the selected points from the model, shown in black spheres, to do the registration.](image)
realigned together and the proposed algorithm is used to calculate the mean-squared value of TRE for all the points on the 3D surface. Figure 6.10 shows the estimated mean-squared value of TRE after performing the registration for each point on the 3D surface using the proposed algorithm. The estimated results are compared with the ones obtained by numerical simulations in 1,000 trials. Table 6.5 shows mean and variance of the estimation error of the mean-squared value of TRE and mean TRE at every point of the 3D surface model. As shown, the proposed algorithm is able to accurately estimate the mean and mean-squared value of TRE at any desired point on the bone surface.

Figure 6.10: Estimated mean-squared value of TRE (in $mm^2$) at every point on the 3D surface model, generated from the scaphoid bone phantom, after registration (please notice the short range of error).
f) Simulation results on a shoulder bone phantom:

In the last experiment, we perform the same simulation as the previous ones with this difference that the data sets are generated from a shoulder bone phantom. The resulting CT images taken from the phantom are segmented semi-automatically by using the software developed at Queen’s University to produce a 3D mesh model containing 138004 points. The fixed data set $Y$ is produced by randomly selecting 30 points from the shoulder surface model, as shown in Figure 6.11. The moving data set $U$ is then generated by applying a random transformation on the chosen 30 points from the 3D mesh and adding zero-mean Gaussian noise, modelling the FLE with covariance matrix $3 \text{mm}^2 I_{3 \times 3}$.

Figure 6.11: 3D surface model of the shoulder bone phantom and the selected points from the model, shown in black spheres, to do the registration.

As before, 1,000 random transformations are generated to take the moving data set out of alignment from the fixed data set. Then, the moving data set is aligned back to the fixed data set, and the proposed algorithm is utilized to calculate the
mean-squared value of TRE at every point of the 3D surface model. Figure 6.12 shows the estimated mean-squared value of TRE, after registration, for each point on the mesh surface using the proposed algorithm. The results are compared with the ones obtained by numerical simulations in 1,000 trails. Table 6.5 shows mean and variance of the estimation error of the mean-squared value of TRE and mean TRE at every point of the shoulder mesh surface. As shown, the proposed algorithm is able to accurately estimate the mean and mean-squared value of TRE at any desired point on the shoulder mesh surface.

![Figure 6.12: Estimated mean-squared value of TRE (in \(mm^2\)) at every point of the 3D surface model of the shoulder bone phantom.](image)

### 6.2 Arbitrary FLE Distribution

Previously, the distribution of TRE was estimated based on the assumption that FLE has an independent identical isotropic distribution for each point in the registering data sets. However, in real clinical applications, due to the properties of medical
Table 6.5: Mean (mm) and variance (mm$^2$) of error of the estimated mean and mean-squared value of TRE.

<table>
<thead>
<tr>
<th>Simulations</th>
<th>E[TRE]</th>
<th>E[TRE$^2$]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Variance</td>
</tr>
<tr>
<td>Pelvic bone</td>
<td>-0.32</td>
<td>0.01</td>
</tr>
<tr>
<td>Scaphoid bone</td>
<td>-0.022</td>
<td>$2 \times 10^{-4}$</td>
</tr>
<tr>
<td>Shoulder bone</td>
<td>-0.032</td>
<td>$4 \times 10^{-5}$</td>
</tr>
</tbody>
</table>

Instruments used for collecting data sets, FLE may have different characteristics and distributions for each point in the data sets. For example, in ultrasound imaging techniques, the variance of the noise in the direction orthogonal to the ultrasound image is greater than those in the ultrasound image plane (noise is anisotropic). Furthermore, since the ultrasound beam-width (signal) may change from a pixel to a pixel in ultrasound image, each pixel may have different noise distributions (noise is inhomogeneous).

In this section, we present a general solution based on the ML algorithm to estimate the distribution of TRE for the cases where FLE has an independent, identical or inhomogeneous, isotropic or anisotropic, distribution at each point in the registering data sets by applying an algorithm that is capable of calculating the optimum registration to the first order. We also show that the proposed algorithm converges to previous art’s results for special cases where FLE has a Gaussian distribution.

### 6.2.1 Maximum Likelihood Algorithm

Here, the goal is to determine a rotation matrix $R(\theta_x, \theta_y, \theta_z)$ and a translation vector $t = [t_x, t_y, t_z]^T$ as well as their covariance matrices from (4.7). In (4.7), $n_i = [n_i^x, n_i^y, n_i^z]^T$. 

\[ R = \begin{pmatrix} c(\theta_z)c(\theta_y) & c(\theta_z)s(\theta_y)s(\theta_x) - s(\theta_z)c(\theta_x) & c(\theta_z)s(\theta_y)c(\theta_x) + s(\theta_z)s(\theta_x) \\ s(\theta_z)c(\theta_y) & s(\theta_z)s(\theta_y)s(\theta_x) + c(\theta_z)c(\theta_x) & s(\theta_z)s(\theta_y)c(\theta_x) - c(\theta_z)s(\theta_x) \\ -s(\theta_y) & c(\theta_y)s(\theta_x) & c(\theta_y)c(\theta_x) \end{pmatrix}, \approx \begin{pmatrix} 1 - \theta^2_y/2 - \theta^2_z/2 & \theta_y\theta_x - \theta_z & \theta_y + \theta_z\theta_x \\ \theta_z & 1 + \theta_z\theta_y\theta_x - \theta^2_z/2 - \theta^2_x/2 & \theta_z\theta_y - \theta_x \\ -\theta_y & 1 - \theta^2_x/2 - \theta^2_y/2 \end{pmatrix} \]

\[
\approx \begin{pmatrix} 0 & -\theta_y & \theta_x \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{pmatrix} + O(\theta^2)
\]

(6.50)

is a zero-mean random vector with covariance matrix \( \Sigma_i^{\hat{y}} \) that models the FLE in localizing the \( i \)th corresponding points in the registering data sets. This means FLE can have an inhomogeneous distribution for each individual point in the data sets. Also, FLE can be isotropic or anisotropic depending on the covariance matrix \( \Sigma_i^{\hat{y}} \). If \( \Sigma_i^{\hat{y}} = \sigma^2_i I_{3 \times 3} \) (\( \sigma^2_i \) is the variance of FLE along each orthogonal axis for the \( i \)’th pair of corresponding points), then FLE is isotropic, but its distribution can still differ among points in the data sets.

To linearize (4.1), we assume that the rotation angles \( \theta_x, \theta_y, \) and \( \theta_z \) are small values (\( \theta_x \gg \theta^2_x, \theta_y \gg \theta^2_y, \) and \( \theta_z \gg \theta^2_z \)) such that \( R \) can be approximated using the first-order Taylor series expansion as shown in (6.50) (in this equation, \( c() \) and \( s() \) represent \( \cos() \) and \( \sin() \), respectively). This assumption is equivalent to the one made by Fitzpatrick and West in (6.4), and is valid as long as the second- and the higher-order Taylor series expansions of the rotation parameters are negligible. Now, we use the ML algorithm to estimate the rotation and translation parameters as well.
as their variances. To do so, one needs to write the log-likelihood function as
\[
\log(P(Y|U, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)) = \log \prod_{i=1}^{N} P(y_i|u_i, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z). \tag{6.51}
\]
For special cases where FLE has a zero-mean Gaussian distribution, then (6.51) is simplified to
\[
\log \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi|\Sigma^\theta_i|}} \exp\left(- \frac{[y_i - Ru_i - t]^T (\Sigma^\theta_i)^{-1} [y_i - Ru_i - t]}{2}\right) = \\
\sum_{i=1}^{N} \log \frac{1}{\sqrt{2\pi|\Sigma^\theta_i|}} + \sum_{i=1}^{N} \frac{[y_i - Ru_i - t]^T (\Sigma^\theta_i)^{-1} [y_i - Ru_i - t]}{2}. \tag{6.52}
\]
To find the rotation and translation parameters that maximize (6.51), one should solve the following equations
\[
d_t = \frac{\partial \log(P(Y|U, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z))}{\partial ([t_x, t_y, t_z])} = 0, \tag{6.53}
\]
\[
d_\theta = \frac{\partial \log(P(Y|U, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z))}{\partial ([\theta_x, \theta_y, \theta_z])} = 0. \tag{6.54}
\]
For the cases where FLE has a zero-mean Gaussian distribution, \(d_t\) and \(d_\theta\) can be analytically computed as follows:
\[
d_t = \sum_{i=1}^{N} (\Sigma^\theta_i)^{-1} \begin{pmatrix} y_i^x - u_i^x + \theta_z u_i^y - \theta_y u_i^z - t_x \\ y_i^y - \theta_z u_i^x - u_i^y + \theta_x u_i^z - t_y \\ y_i^z + \theta_y u_i^x - \theta_x u_i^y - u_i^z - t_z \end{pmatrix}, \tag{6.55}
\]
\[
d_\theta = -\sum_{i=1}^{N} \begin{pmatrix} u_i^y & -u_i^z & 0 \\ -u_i^z & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix} (\Sigma^\theta_i)^{-1} \begin{pmatrix} y_i^x - u_i^x + \theta_z u_i^y - \theta_y u_i^z - t_x \\ y_i^y - \theta_z u_i^x - u_i^y + \theta_x u_i^z - t_y \\ y_i^z + \theta_y u_i^x - \theta_x u_i^y - u_i^z - t_z \end{pmatrix}. \tag{6.56}
\]
The registration parameters can be computed simultaneously by concurrently solving (6.53) and (6.54).
For example, when FLE has an identical isotropic zero-mean Gaussian distribution, \( \mathcal{N}(0, \sigma^2 I) \), by these assumptions that the centroid of \( \mathbf{X} \) is set to the origin, \( \bar{x} = 0 \), and the axes of \( \mathbf{X} \) are realigned with its orthogonal eigenvectors, the registration parameters, \( \hat{\mathbf{t}} = [t_x, t_y, t_z]^T \) and \( \hat{\mathbf{\theta}} = [\theta_x, \theta_y, \theta_z]^T \), can be easily computed as

\[
\hat{\mathbf{t}} = \begin{pmatrix} \hat{t}_x \\ \hat{t}_y \\ \hat{t}_z \end{pmatrix} = \frac{1}{N} \sum_{i=1}^{N} \begin{pmatrix} n_i^x \\ n_i^y \\ n_i^z \end{pmatrix}, \tag{6.57}
\]

and

\[
\hat{\mathbf{\theta}} = \begin{pmatrix} \hat{\theta}_x \\ \hat{\theta}_y \\ \hat{\theta}_z \end{pmatrix} = \sum_{i=1}^{N} \begin{pmatrix} -u_i^y y_i^x + u_i^x y_i^y \\ \sum_{i=1}^{N} (u_i^y)^2 + \sum_{i=1}^{N} (u_i^x)^2 \\ u_i^z y_i^x - u_i^x y_i^z \\ \sum_{i=1}^{N} (u_i^z)^2 + \sum_{i=1}^{N} (u_i^x)^2 \\ -u_i^z y_i^y + u_i^y y_i^z \\ \sum_{i=1}^{N} (u_i^y)^2 + \sum_{i=1}^{N} (u_i^z)^2 \end{pmatrix}, \tag{6.58}
\]

which respectively match the ones calculated by Fitzpatrick and West (\( i.e., \) Equations (6.3) and (6.6)).

To estimate the variance of the registration parameters, one needs to derive the second derivative of the log-likelihood function, Equation (6.51), as follows:

\[
\mathbf{J}_{tt} = \frac{\partial^2 \log(P(\mathbf{Y}|\mathbf{U}, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z))}{\partial([t_x, t_y, t_z]) \partial([t_x, t_y, t_z])}, \tag{6.59}
\]

\[
\mathbf{J}_{t\theta} = \frac{\partial^2 \log(P(\mathbf{Y}|\mathbf{U}, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z))}{\partial([t_x, t_y, t_z]) \partial([\theta_x, \theta_y, \theta_z])}, \tag{6.60}
\]

\[
\mathbf{J}_{\theta\theta} = \frac{\partial^2 \log(P(\mathbf{Y}|\mathbf{U}, t_x, t_y, t_z, \theta_x, \theta_y, \theta_z))}{\partial([\theta_x, \theta_y, \theta_z]) \partial([\theta_x, \theta_y, \theta_z])}. \tag{6.61}
\]

For the cases where FLE has a zero-mean Gaussian distribution \( \mathbf{J}_{tt}, \mathbf{J}_{t\theta} \) and \( \mathbf{J}_{\theta\theta} \) can
be respectively computed as

\[ J_{tt} = - \sum_{i=1}^{N} (\Sigma_i^{\hat{y}})^{-1}, \]  

(6.62)

\[ J_{t\theta} = \sum_{i=1}^{N} (\Sigma_i^{\hat{y}})^{-1} \begin{pmatrix} u_i^y - u_i^x & 0 \\ -u_i^z & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix}, \]  

(6.63)

\[ J_{\theta\theta} = - \sum_{i=1}^{N} \left[ \begin{pmatrix} u_i^y \\ -u_i^x \\ -u_i^z \\ 0 \end{pmatrix} (\Sigma_i^{\hat{y}})^{-1} \begin{pmatrix} u_i^y \\ -u_i^x \\ -u_i^z \\ 0 \end{pmatrix} \right]^T \]  

(6.64)

Finally, by using the Cramer-Rao bound [123], covariance matrix of the registration parameters can be computed as

\[ \Sigma = \mathbb{E}[\left( t_x - \hat{t}_x, t_y - \hat{t}_y, t_z - \hat{t}_z, \theta_z - \hat{\theta}_z, \theta_y - \hat{\theta}_y, \theta_x - \hat{\theta}_x \right)^2] \]

\[ = \begin{pmatrix} \Sigma_{xt} & \Sigma_{xt\theta} \\ \Sigma_{xt\theta} & \Sigma_{\theta \theta} \end{pmatrix} = J^{-1} \begin{pmatrix} J_{tt} & -J_{t\theta} \\ -J_{t\theta}^T & J_{\theta\theta} \end{pmatrix}^{-1}, \]  

(6.65)

where \( \Sigma_{xt} \) and \( \Sigma_{\theta \theta} \), as defined earlier, are covariance matrices of the translation and rotation parameters, respectively, and \( \Sigma_{xt\theta} \) is the covariance matrix among the translation and rotation parameters.

Equation (6.65) can be analytically calculated for special cases where FLE has identical or inhomogeneous isotropic zero-mean Gaussian distributions:

1) For the cases when FLE has an identical isotropic zero-mean Gaussian distribution, \( \Sigma_i^{\hat{y}} \) has the same value at each point of the data sets, and along each orthogonal

\[ 2 \text{ A simple proof of (6.65) is given in Appendix C.} \]
axis, i.e., $\Sigma_i^g = \sigma^2 I$. Therefore, by assuming that the center of the data sets is at the origin and that the orientation of the data sets is along their eigenvectors, then (6.62), (6.63), and (6.64) can be simplified, respectively, as

$$J_{tt} = -\sum_{i=1}^{N} (\sigma^2 I)^{-1} = -\frac{N}{\sigma^2},$$  \hspace{1cm} (6.66)

$$J_{t\theta} = \frac{1}{\sigma^2} \sum_{i=1}^{N} \begin{pmatrix} u_i^y & -u_i^x & 0 \\ -u_i^z & 0 & u_i^z \\ 0 & u_i^z & -u_i^y \end{pmatrix} = 0,$$  \hspace{1cm} (6.67)

$$J_{\theta\theta} = -\frac{1}{\sigma^2} \sum_{i=1}^{N} \begin{pmatrix} (u_i^y)^2 + (u_i^x)^2 & -u_i^y u_i^z & -u_i^x u_i^z \\ -u_i^y u_i^z & (u_i^z)^2 + (u_i^x)^2 & -u_i^z u_i^y \\ -u_i^x u_i^z & -u_i^z u_i^y & (u_i^y)^2 + (u_i^x)^2 \end{pmatrix}^T,$$  \hspace{1cm} (6.68)

$$= -\frac{1}{\sigma^2} \sum_{i=1}^{N} \begin{pmatrix} (u_i^y)^2 + (u_i^x)^2 & 0 & 0 \\ 0 & (u_i^z)^2 + (u_i^x)^2 & 0 \\ 0 & 0 & (u_i^z)^2 + (u_i^y)^2 \end{pmatrix}^T.$$  \hspace{1cm}

By substituting (6.66), (6.67), and (6.68) in (6.65), we have

$$\Sigma_{xt} = \frac{\sigma^2}{N} I,$$  \hspace{1cm} (6.69)

$$\Sigma_{xtx_0} = 0,$$  \hspace{1cm} (6.70)
Equations (6.69), (6.70), and (6.71) show that the translational and rotational parameters are uncorrelated, and their variances have direct and inverse relations with the variance of FLE and number of fiducials in the registering data sets, respectively.

2) If FLE has an inhomogeneous isotropic zero-mean Gaussian distribution in the data sets, then $\Sigma_i^g$ has the same value along each orthogonal axis but its value is different from one point to another in the data sets, i.e., $\Sigma_i^g = \sigma_i^2 I$. In this case, $J_{tt}$, $J_{t\theta}$ and $J_{\theta\theta}$ can be analytically computed from (6.62), (6.63), and (6.64), respectively, as

$$J_{tt} = -\sum_{i=1}^{N} \frac{1}{\sigma_i^2} I,$$

$$J_{t\theta} = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \begin{pmatrix} u_i^y & -u_i^x & 0 \\ -u_i^z & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix},$$

$$J_{\theta\theta} = \sum_{i=1}^{N} \frac{-1}{\sigma_i^2} \begin{pmatrix} (u_i^y)^2 + (u_i^x)^2 & -u_i^y u_i^z & -u_i^x u_i^z \\ -u_i^y u_i^z & (u_i^z)^2 + (u_i^x)^2 & -u_i^y u_i^x \\ -u_i^x u_i^z & -u_i^y u_i^x & (u_i^z)^2 + (u_i^y)^2 \end{pmatrix}.$$

Now, if the weighted mean of data set $U$ is assumed to be at the origin, i.e., $\bar{u} = \bar{y} = \bar{z} = 0$, then
CHAPTER 6. MEASUREMENT OF THE REGISTRATION ACCURACY

\[ \sum_{i=1}^{N} \frac{u_i}{\sigma_i^2} = 0, \quad \text{then} \]
\[ \frac{1}{N} \sum_{i=1}^{N} \sigma_i^2 = 0, \quad \text{then} \quad J_{t\theta} = 0. \]  
(6.75)

Also, if the orientation of data set \( U \) is chosen such that \( \sum_{i=1}^{N} \frac{u_x u_y}{\sigma_i^2} = \sum_{i=1}^{N} \frac{u_x u_z}{\sigma_i^2} = \sum_{i=1}^{N} \frac{u_y u_z}{\sigma_i^2} = 0 \), then \( J_{\theta\theta} \) simplifies to
\[
J_{\theta\theta} = -\sum_{i=1}^{N} \begin{pmatrix}
\frac{(u_y^i)^2 + (u_z^i)^2}{\sigma_i^2} & 0 & 0 \\
0 & \frac{(u_z^i)^2 + (u_x^i)^2}{\sigma_i^2} & 0 \\
0 & 0 & \frac{(u_x^i)^2 + (u_y^i)^2}{\sigma_i^2}
\end{pmatrix}.
\]  
(6.76)

By substituting (6.72), (6.75), and (6.76) into (6.65), \( \Sigma_{x_t} \), \( \Sigma_{x_t x_\theta} \), and \( \Sigma_{x_\theta} \) of covariance matrix \( \Sigma \) can be calculated as follows:
\[
\Sigma_{x_t} = \frac{1}{N}, \quad \sum_{i=1}^{N} \frac{1}{\sigma_i^2}, \quad \Sigma_{x_t x_\theta} = 0, \quad \Sigma_{x_\theta} = \begin{pmatrix}
\frac{1}{N} \sum_{i=1}^{N} \frac{(u_y^i)^2 + (u_x^i)^2}{\sigma_i^2} & 0 & 0 \\
0 & \frac{1}{N} \sum_{i=1}^{N} \frac{(u_z^i)^2 + (u_x^i)^2}{\sigma_i^2} & 0 \\
0 & 0 & \frac{1}{N} \sum_{i=1}^{N} \frac{(u_x^i)^2 + (u_y^i)^2}{\sigma_i^2}
\end{pmatrix}.
\]  
(6.77, 6.78, 6.79)

Equations (6.77), (6.78), and (6.79) show that the translational and rotational parameters, as before, are uncorrelated, and their variances have direct and inverse
relations with the variances of FLE and number of fiducials in the registering data sets, respectively.

3) For the cases where FLE has an identical or inhomogeneous anisotropic zero-mean Gaussian distribution, Equations (6.62), (6.63), and (6.64) cannot be analytically solved, and therefore, one needs to numerically calculate those equations, and compute the result from (6.65).

Now that the registration parameters and their variances are estimated, in what follows, we explain how to use those variances to estimate the distribution of TRE and its mean-squared value at a target location.

a) Estimating the distribution of TRE:

Let us assume that the registration is performed between two data sets, and the two data sets are nearly aligned such that the approximation used in (6.50) is valid. Now, let \( \mathbf{r} \) be a point at which we wish to calculate TRE. To derive the distribution of TRE at the target location \( \mathbf{r} \), as before, we approximate the random vector \( \mathbf{e}_r = [e_x, e_y, e_z]^T \) in (6.25) by a zero-mean Gaussian random vector with covariance matrix \( \Sigma_e \), as defined in (6.32).

To analytically calculate \( \Sigma_e \) in terms of \( \Sigma \), we respectively calculate \( \Sigma_{\mathbf{R}_r} \), \( \Sigma_{(\mathbf{R}_r)^t} \), \( \Sigma_{\mathbf{t}(\mathbf{R}_r)} \), and \( \Sigma_{\mathbf{x}_t} \) in (6.32), and sum them up. By using (6.50), \( \Sigma_{\mathbf{R}_r} \) can be computed as

\[
\Sigma_{\mathbf{R}_r} = \begin{pmatrix}
\Sigma_{11}^{\mathbf{R}_r} & \Sigma_{12}^{\mathbf{R}_r} & \Sigma_{13}^{\mathbf{R}_r} \\
\Sigma_{12}^{\mathbf{R}_r} & \Sigma_{22}^{\mathbf{R}_r} & \Sigma_{23}^{\mathbf{R}_r} \\
\Sigma_{13}^{\mathbf{R}_r} & \Sigma_{23}^{\mathbf{R}_r} & \Sigma_{33}^{\mathbf{R}_r}
\end{pmatrix} = \mathbf{E}\left[\begin{pmatrix}
\Sigma_{11}^{\mathbf{R}_r} & \Sigma_{12}^{\mathbf{R}_r} & \Sigma_{13}^{\mathbf{R}_r} \\
\Sigma_{12}^{\mathbf{R}_r} & \Sigma_{22}^{\mathbf{R}_r} & \Sigma_{23}^{\mathbf{R}_r} \\
\Sigma_{13}^{\mathbf{R}_r} & \Sigma_{23}^{\mathbf{R}_r} & \Sigma_{33}^{\mathbf{R}_r}
\end{pmatrix}\right]
\]
where

\[
\begin{align*}
\Sigma_{11}^{Rr} &= r_y^2 \Sigma_{xx} - 2r_y r_z \Sigma_{xz} + r_z^2 \Sigma_{zz}, \\
\Sigma_{12}^{Rr} &= -r_x r_y \Sigma_{xz} + r_y r_z \Sigma_{xz} + r_x r_z \Sigma_{xz} - r_z^2 \Sigma_{x0}, \\
\Sigma_{13}^{Rr} &= r_x r_y \Sigma_{xz} + r_y r_z \Sigma_{xz} - r_x r_z \Sigma_{xz} - r_z^2 \Sigma_{x0}, \\
\Sigma_{22}^{Rr} &= -r_x^2 \Sigma_{xz} + r_x r_z \Sigma_{xz} + r_x r_z \Sigma_{xz} + r_y r_z \Sigma_{xz}, \\
\Sigma_{33}^{Rr} &= -2r_x r_y \Sigma_{xz} - 2r_y r_z \Sigma_{xz} + r_y^2 \Sigma_{xx}.
\end{align*}
\]

Following the same way, \( \Sigma_{(Rr)t} \) can be computed as

\[
E\left[ \begin{pmatrix} 0 & \hat{\theta}_z - \theta_z & \theta_y - \hat{\theta}_y \\ \theta_z - \hat{\theta}_z & 0 & \hat{\theta}_x - \theta_x \\ \hat{\theta}_y - \theta_y & \hat{\theta}_x - \theta_x & 0 \end{pmatrix} \right] \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} \times \begin{pmatrix} t_x - \hat{t}_x \\ t_y - \hat{t}_y \\ t_z - \hat{t}_z \end{pmatrix},
\]

which can be simplified to

\[
\Sigma_{(Rr)t} = \begin{pmatrix} -r_y \Sigma_{xx}^{11} + r_z \Sigma_{xz}^{11} - r_y \Sigma_{xx}^{12} + r_z \Sigma_{xz}^{12} - r_y \Sigma_{xx}^{31} + r_z \Sigma_{xz}^{32} \\ -r_x \Sigma_{xx}^{13} + r_y \Sigma_{xx}^{11} - r_x \Sigma_{xx}^{12} + r_y \Sigma_{xx}^{21} - r_x \Sigma_{xx}^{31} + r_y \Sigma_{xx}^{33} \\ -r_x \Sigma_{xx}^{13} + r_y \Sigma_{xx}^{11} - r_x \Sigma_{xx}^{12} + r_y \Sigma_{xx}^{21} - r_x \Sigma_{xx}^{31} + r_y \Sigma_{xx}^{33} \end{pmatrix},
\]

(6.81)

where \( \Sigma_{xx}^{ij} \) is the \( ij \) element of covariance matrix \( \Sigma_{xx} \).

\( \Sigma_{t(Rr)} \) can be computed from \( \Sigma_{(Rr)t} \) as

\[
\Sigma_{t(Rr)} = \Sigma_{(Rr)t}^T,
\]

(6.82)
Finally, $\Sigma_{er}$ can be computed in a general form by substituting (6.80), (6.81), (6.82), and (6.83) in (6.32).

For a special case, when FLE has an identical or inhomogeneous isotropic zero-mean Gaussian distribution, since the rotation and translation parameters become uncorrelated, (6.80), (6.81), and (6.83) can be simplified to

$$\Sigma_{(Rr)^t} = \Sigma_{t(Rr)} = 0,$$  \hspace{1cm} (6.85)

\[
\Sigma_{xt} = \begin{pmatrix}
\Sigma_{11}^{xt} & 0 & 0 \\
0 & \Sigma_{22}^{xt} & 0 \\
0 & 0 & \Sigma_{33}^{xt}
\end{pmatrix}.
\hspace{1cm} (6.86)
\]

Finally, $\Sigma_{er}$ can be calculated by adding (6.84), (6.85), and (6.86) as

\[
\Sigma_{er} = \begin{pmatrix}
y^2\Sigma_{11}^{xt} + y^2\Sigma_{11}^{xt} + \Sigma_{11}^{xt} & -y^2r^{yt}\Sigma_{11}^{xt} & -y^2r^{zt}\Sigma_{11}^{xt} \\
-y^2r^{yt}\Sigma_{11}^{xt} & r^2\Sigma_{11}^{xt} + r^2\Sigma_{11}^{xt} + \Sigma_{22}^{xt} & -y^2r^{zt}\Sigma_{11}^{xt} \\
-y^2r^{zt}\Sigma_{11}^{xt} & -y^2r^{zt}\Sigma_{11}^{xt} & r^2\Sigma_{22}^{xt} + r^2\Sigma_{22}^{xt} + \Sigma_{33}^{xt}
\end{pmatrix}.
\hspace{1cm} (6.87)
\]

Equation (6.87) shows that $e_x, e_y$, and $e_z$ are three correlated zero-mean Gaussian random variables with covariance matrix $\Sigma_{er}$. As mentioned before, since TRE is a distance, and therefore, independent to rotations, one can transform the three correlated Gaussian random variables to three independent (uncorrelated) Gaussian
random variables by applying a rotation matrix to the error vector \( e(r) \). This rotation matrix is the eigenvectors of covariance matrix \( \Sigma_e \). Then, TRE can be reformulated as square root of summation of three squared independent Gaussian random variables with variances \( \sigma^2_{nx}, \sigma^2_{ny} \) and \( \sigma^2_{nz} \) which are the eigenvalues of \( \Sigma_{e}r \). The variances can then be used in (6.41) to derive the distributions of TRE at the target location \( r \).

b) Estimating the mean-squared value of TRE:

By assuming that the registration between the two data sets is performed and (6.50) is valid, we can use (6.42) to analytically compute the mean-squared value of TRE at the target point \( r \).

\[
E[\Delta t^T \Delta t], \text{ the last term of (6.42), can be estimated as }
E[\Delta t^T \Delta t] = E[(t_x - \hat{t}_x)^2] + E[(t_y - \hat{t}_y)^2] + E[(t_z - \hat{t}_z)^2] = \Sigma_{xt}^{x} + \Sigma_{xt}^{y} + \Sigma_{xt}^{z}, \tag{6.88}
\]

\[
E[\Delta R^T \Delta t], \text{ the second part of (6.42), can be written as }
E[\Delta R^T \Delta t] = E[
\begin{pmatrix}
(\theta_x - \hat{\theta}_x)(t_y - \hat{t}_y)
& (\theta_y - \hat{\theta}_y)(t_z - \hat{t}_z)
& (\theta_z - \hat{\theta}_z)(t_x - \hat{t}_x)
\end{pmatrix}
\begin{pmatrix}
\Sigma_{x\theta x}^{y} - \Sigma_{x\theta x}^{y}
\Sigma_{x\theta x}^{y} + \Sigma_{x\theta x}^{y} - \Sigma_{x\theta x}^{y}
\Sigma_{x\theta x}^{z} - \Sigma_{x\theta x}^{z}
\end{pmatrix}, \tag{6.89}
\]

where \( \Sigma_{x\theta x}^{ij} \) is the \( ij \)th element of covariance matrix \( \Sigma_{t\theta} \) in (6.65). Finally, \( E[\Delta R^T \Delta R] \), the first part of (6.42), can be computed as

\[
E[\Delta R^T \Delta R] =
\begin{pmatrix}
\Sigma_{x\theta x}^{11} + \Sigma_{x\theta x}^{22} & -\Sigma_{x\theta x}^{23} & -\Sigma_{x\theta x}^{13}
-\Sigma_{x\theta x}^{23} & \Sigma_{x\theta x}^{11} + \Sigma_{x\theta x}^{33} & -\Sigma_{x\theta x}^{12}
-\Sigma_{x\theta x}^{13} & -\Sigma_{x\theta x}^{12} & \Sigma_{x\theta x}^{22} + \Sigma_{x\theta x}^{33}
\end{pmatrix}. \tag{6.90}
\]

By substituting (6.88), (6.89), and (6.90) into (6.42), the mean-squared value of TRE at the target location \( r \) can be calculated for a general case where FLE has an arbitrary
zero-mean distribution in the data sets, as follows:

\[
E[TRE_r^2] = r^T \begin{pmatrix}
\Sigma_{x_0}^{11} + \Sigma_{x_0}^{22} & -\Sigma_{x_0}^{23} & -\Sigma_{x_0}^{13} \\
-\Sigma_{x_0}^{23} & \Sigma_{x_0}^{11} + \Sigma_{x_0}^{33} & -\Sigma_{x_0}^{12} \\
-\Sigma_{x_0}^{13} & -\Sigma_{x_0}^{12} & \Sigma_{x_0}^{22} + \Sigma_{x_0}^{33}
\end{pmatrix} \Sigma_{x_0}^{22} r + 2 \begin{pmatrix}
\Sigma_{x_0}^{21} - \Sigma_{x_0}^{32} \\
\Sigma_{x_0}^{33} - \Sigma_{x_0}^{11} \\
\Sigma_{x_0}^{12} - \Sigma_{x_0}^{23}
\end{pmatrix}^T r
+ \Sigma_{x_t}^{11} + \Sigma_{x_t}^{22} + \Sigma_{x_t}^{33}.
\]

(6.91)

In what follows, we simplify (6.91) for special cases where FLE has an identical or inhomogeneous isotropic zero-mean Gaussian distributions.

1) If FLE has an identical isotropic zero-mean Gaussian distribution at each point of the data sets \(\Sigma_{i}^{y} = \sigma_{i}^2 \mathbf{I}_{3 \times 3}\), by using (6.69), (6.70), and (6.71) in (6.91), we have

\[
E[TRE_r^2] = \frac{3\sigma^2}{N} + r_x^2 \left( \frac{\sigma^2}{\sum_{i=1}^{N}(u_{i}^{y})^2 + \sum_{i=1}^{N}(u_{i}^{x})^2} \right) + \frac{\sigma^2}{\sum_{i=1}^{N}(u_{i}^{z})^2 + \sum_{i=1}^{N}(u_{i}^{x})^2} \right) + \frac{\sigma^2}{\sum_{i=1}^{N}(u_{i}^{z})^2 + \sum_{i=1}^{N}(u_{i}^{y})^2}
\]

(6.92)

Equation (6.92) exactly matches the formula derived by Fitzpatrick and West [36], \textit{i.e.}, Equation (6.23).

2) If FLE has an inhomogeneous isotropic zero-mean Gaussian noise distribution, then (6.77), (6.78), and (6.79) can be used in (6.91) to compute the mean-squared
value of TRE at the target location \( r \), as

\[
E[TRE_r^2] = \frac{3}{N} \sum_{i=1}^{N} \left( \frac{1}{\sigma_i^2} \right) + \sum_{i=1}^{N} \left( \frac{1}{\sigma_i^2} \right) \left( \frac{u_i^y}{\sigma_i^2} \right)^2 + \sum_{i=1}^{N} \left( \frac{1}{\sigma_i^2} \right) \left( \frac{u_i^x}{\sigma_i^2} \right)^2 + \sum_{i=1}^{N} \left( \frac{1}{\sigma_i^2} \right) \left( \frac{u_i^z}{\sigma_i^2} \right)^2
\]

(6.93)

3) Finally, for the cases where FLE has an identical or inhomogeneous anisotropic zero-mean Gaussian distribution at each point of the data sets, the estimated rotation and translation parameters are no longer uncorrelated (i.e., \( \Sigma_{t\theta} \neq 0 \)) and \( J_{tt} \), \( J_{t\theta} \), and \( J_{\theta\theta} \) cannot be analytically calculated. Therefore, \( \Sigma \) cannot be simply computed as before, and should be numerically computed. \( \Sigma \) can then be used in (6.91) to numerically compute the mean-squared value of TRE.

Later on, by using numerical simulations in Section 6.2.5, we will verify the presented formulae in order to calculate the mean-squared value of TRE for different types of FLE distribution in the data sets. But, before moving to that section, we would like to illustrate that the previous TRE estimators presented in the literature, such as Sielhorst [114], Ma [71], and Wiles [133] algorithms, are special cases of our proposed method.

### 6.2.2 Sielhorst Algorithm

Sielhorst et al. [114] employed the transform of covariance algorithm, introduced by Hoff and Vincent [46], to calculate the covariance matrix of the transformation
parameters in the presence of inhomogeneous and anisotropic zero-mean Gaussian FLE. The authors defined covariance matrix of the transformation parameters, \( \Sigma_s \), as follows:

\[
\Sigma_s = \left( M^T \begin{bmatrix} \Sigma_{y} & 0 \\ \cdot & \cdot \\ 0 & \Sigma_{y} \end{bmatrix}^{-1} M \right)^{-1},
\]

(6.94)

where \( M \) is a concatenation of Jacobian matrix of (4.1) calculated at each point in the data set \( X \), as

\[
M = [J(u_1)^T, ..., J(u_N)^T]^T, \quad J(u_i) = \begin{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -u_{iy}^i & u_{iz}^i \\ 0 & 1 & 0 & u_{ix}^i \\ 0 & 0 & 1 & 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \cdot \\ 0 \end{bmatrix} \end{bmatrix} = (I - B_i^T),
\]

(6.95)

where \( B_i \) is defined in (6.63). By substituting \( J(u_i) \) into \( M \), and \( M \) into (6.94), \( \Sigma_s \) can be written as

\[
\Sigma_s = \left( \begin{bmatrix} \begin{bmatrix} I & & \\ & \cdots & \\ & & I \end{bmatrix} & \begin{bmatrix} (\Sigma_{y1})^{-1} & 0 \\ \cdot & \cdot \\ 0 & (\Sigma_{yN})^{-1} \end{bmatrix} \end{bmatrix}^{-1} \begin{bmatrix} I & -B_1^T \\ \cdot & \cdot \\ I & -B_N^T \end{bmatrix} \right)^{-1}
\]

\[
= \left( \begin{bmatrix} \begin{bmatrix} I & & \\ & \cdots & \\ & & I \end{bmatrix} & \begin{bmatrix} (\Sigma_{y1})^{-1} & -B_1^T & \\ \cdot & \cdot & \cdot \\ (\Sigma_{yN})^{-1} & -B_N^T \end{bmatrix} \end{bmatrix}^{-1} \right)^{-1}
\]

\[
= \left( \begin{bmatrix} \begin{bmatrix} \sum_{i=1}^N (\Sigma_{y1})^{-1} & -\sum_{i=1}^N (\Sigma_{y1})^{-1} B_i^T \\ & \cdot \\ & -\sum_{i=1}^N B_i (\Sigma_{y1})^{-1} \end{bmatrix} \end{bmatrix}^{-1} \right)^{-1}.
\]

(6.96)
Comparing (6.96) and (6.65) verifies that both the transform of the covariance and ML algorithms reach the same solution regarding estimation of the covariance matrix of the transformation parameters when FLE has an inhomogeneous and anisotropic zero-mean Gaussian distribution. As explained in Section 6.2.1, (6.96) can then be used to estimate the distribution of TRE and its mean-squared value at the target location $r$.

### 6.2.3 Ma Algorithm

Ma et al. [71] utilized the spatial stiffness theory to estimate the mean-squared value of TRE to within a constant factor, when FLE has an inhomogeneous and anisotropic zero-mean Gaussian distribution with covariance matrix $\Sigma_{i}^{\tilde{y}} = \text{diag}(\frac{1}{k_{x_{i}}}, \frac{1}{k_{y_{i}}}, \frac{1}{k_{z_{i}}})$. Using the spatial stiffness theory, the covariance matrix of the transformation parameters is computed as $\Sigma_{b} = (\sum_{i=1}^{N} \mathbf{H}_{i})^{-1}$, where

$$
\mathbf{H}_{i} = \begin{pmatrix}
    k_{x_{i}} & 0 & 0 & -k_{x_{i}}u_{i}^{y} & k_{x_{i}}u_{i}^{z} & 0 \\
    0 & k_{y_{i}} & 0 & k_{y_{i}}u_{i}^{x} & 0 & -k_{y_{i}}u_{i}^{z} \\
    0 & 0 & k_{z_{i}} & 0 & -k_{z_{i}}u_{i}^{x} & k_{z_{i}}u_{i}^{y} \\
    -k_{x_{i}}u_{i}^{y} & k_{y_{i}}u_{i}^{x} & 0 & k_{x_{i}}(u_{i}^{y})^{2} + k_{y_{i}}(u_{i}^{x})^{2} & -k_{x_{i}}u_{i}^{y}u_{i}^{z} & k_{x_{i}}u_{i}^{y} \\
    k_{x_{i}}u_{i}^{z} & 0 & -k_{x_{i}}u_{i}^{x} & -k_{x_{i}}u_{i}^{y}u_{i}^{z} & k_{x_{i}}(u_{i}^{z})^{2} + k_{x_{i}}(u_{i}^{y})^{2} & -k_{x_{i}}u_{i}^{y}u_{i}^{z} \\
    0 & -k_{y_{i}}u_{i}^{z} & k_{z_{i}}u_{i}^{y} & -k_{y_{i}}u_{i}^{z}u_{i}^{x} & -k_{y_{i}}u_{i}^{z}u_{i}^{y} & k_{y_{i}}(u_{i}^{z})^{2} + k_{z_{i}}(u_{i}^{y})^{2} 
\end{pmatrix},
$$

(6.97)

By substituting (6.97) into $\Sigma_{b}$, it can be easily verified that $\Sigma_{b} = \Sigma_{c} = \Sigma$. Having calculated $\Sigma_{b}$, as explained in Section 6.2.1, one can easily estimate the distribution of TRE and its mean-squared value at the target location $r$. 

6.2.4 Wiles Algorithm

Wiles et al. [133] derived a closed-form solution that estimates the mean-squared value of TRE when FLE has an identical and anisotropic zero-mean Gaussian distribution. Their method, similar to Fitzpatrick and West algorithm, utilizes a least-mean-squares algorithm, such as Horn [47, 48] or Arun [2] algorithm, to estimate the registration parameters and their variances. These authors used (6.57) and (6.58) as the estimated registration parameters and calculated the variances of the registration parameters to compute the mean squared value of TRE. However, since FLE has an anisotropic distribution, a weighted least-mean-squares algorithm could also be used to more accurately estimate the registration parameters and improve their proposed method. By assuming that FLE has an identical anisotropic zero-mean Gaussian distribution, \( N(0, \Sigma^g) \), the registration parameters can be accurately computed from (6.55), and (6.56) as

\[
\hat{t} = \sum_{i=1}^{N} (N)^{-1} \begin{pmatrix} y_i^x - u_i^x \\ y_i^y - u_i^y \\ y_i^z - u_i^z \end{pmatrix},
\]

\[
\hat{\theta} = \frac{-\sum_{i=1}^{N} \begin{pmatrix} u_i^y & -u_i^x & 0 \\ -u_i^y & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix} (\Sigma^g)^{-1} \begin{pmatrix} y_i^x - u_i^x \\ y_i^y - u_i^y \\ y_i^z - u_i^z \end{pmatrix}}{\sum_{i=1}^{N} \begin{pmatrix} u_i^y & -u_i^x & 0 \\ -u_i^y & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix} (\Sigma^g)^{-1} \begin{pmatrix} u_i^y & -u_i^x & 0 \\ -u_i^y & 0 & u_i^x \\ 0 & u_i^z & -u_i^y \end{pmatrix}},
\]
CHAPTER 6. MEASUREMENT OF THE REGISTRATION ACCURACY

which are not necessarily equal to (6.57) and (6.58). Their derived closed-form formula would match our proposed derivations if they would have used (6.98) and (6.99) instead of (6.57) and (6.58) in their formulations. In experimental simulations, we compare the performance of the Wiles and our proposed algorithm by using numerical simulations.

6.2.5 Numerical Simulations

In order to test our approximations and verify our derivations, several numerical simulations on random data sets are performed. First, we run the simulations on the data sets which are perturbed by inhomogeneous isotropic zero-mean Gaussian distributions, to derive the distribution of TRE and mean-squared value of TRE at a target location based on the proposed ML algorithm. In the second simulation, it is shown that, for the cases where FLE has identical isotropic zero-mean Gaussian distributions, the estimated distribution of TRE and its mean-squared value based on ML algorithm match those derived by Fitzpatrick and West algorithm. In the third and fourth simulations, respectively, the performance of the proposed algorithm is examined for the cases where FLE has an identical or inhomogeneous, isotropic or anisotropic zero-mean Gaussian distribution at each point of the registering data sets. Finally, in the last experiment, the performance of the proposed, Sielhorst [114], Ma [71], Wiles [133], and Fitzpatrick and West [36] algorithms are compared when FLE has identical isotropic and inhomogeneous anisotropic zero-mean Gaussian noise, respectively.
a) **FLE has inhomogeneous isotropic zero-mean Gaussian distributions, \( \mathcal{N}(0, \sigma_i^2 I_{3\times3}) \), at each point of the data sets:**

First, we choose \( N \), the number of points in the data sets, to be 10. For this value of \( N \), we randomly generate the moving data set \( U \) within a cube with sides of \( \pm 100 \text{ mm} \) and one target, \( r \), within the cube with the sides of \( \pm 200 \text{ mm} \). To generate the fixed data set \( Y \), we perturb each point in \( U \) by a zero-mean Gaussian random vector with covariance matrix \( \sigma_i^2 I_{3\times3} \). \( \sigma_i^2 \) (\( i = 1, ..., N \)) is uniformly drawn between 0 and 30 \( \text{mm}^2 \). We register the data set \( Y \) to \( U \) using the UKF registration algorithm, and calculate the TRE at \( r \) using (6.2). One simulation consists of 100,000 repetitions of perturbation, registration, and TRE calculation steps which allows us to numerically estimate the distribution of TRE and its mean-squared value. Then, we use the proposed formulae, Equations (6.41) and (6.93), to estimate the distribution of TRE and its mean-squared value, respectively.

Figure 6.13 shows the estimated distributions of TRE and the one generated by numerical simulations at the target point \( r = [-38.4, -13.7, 97.2]^T \), in mm. Also, Table 6.6 displays the estimated mean-squared values of TRE and the distance between the obtained distributions using the proposed algorithm and numerical simulations.

As can be seen from Figure 6.13 and Table 6.6, the results of the proposed algorithm accurately follow those generated by numerical simulations for the cases where FLE has inhomogeneous isotropic zero-mean Gaussian distribution at each point of the data sets.

We repeat this simulation for 100 trials with a variety of randomly selected fiducial configurations and target locations. In each trial, we compare the root mean-squared value of TRE and its distribution estimated by the proposed algorithm with the ones
obtained by numerical simulations. Table 6.10 displays the mean and variance of error difference between the calculated root mean-squared values of TRE using the proposed algorithm and numerical simulations, and the mean distance between the two estimated distributions over 100 trials, when $N$ (number of fiducials) is 3, 4, 10, 20, and 50, respectively. As shown in Table 6.10, the proposed algorithm accurately estimates the TRE distribution and its mean-squared value when FLE has inhomogeneous isotropic zero-mean Gaussian distribution at each point of the data sets.

Figure 6.13: Probability densities of the estimated target registration error at $r$ using numerical simulations and the proposed algorithm when the data sets are perturbed by inhomogeneous isotropic zero-mean Gaussian FLE.

b) **FLE has an identical isotropic zero-mean Gaussian distribution,** $\mathcal{N}(0, \sigma^2 I_{3\times3})$, at each point of the data sets:

As in the previous simulation, the data set $U$ and target $r$ are randomly generated. But, to generate the fixed data set $Y$, each point in $U$ is perturbed by identical
Table 6.6: Estimated mean-squared values of TRE and distance between estimated distributions of TRE at \( r \) using numerical simulations and the proposed algorithm when the data sets are perturbed by inhomogeneous isotropic zero-mean Gaussian FLE.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \mathbb{E}<a href="mm%5E2">\text{TRE}^2</a> )</th>
<th>( d(pdf_1, pdf_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical simulations</td>
<td>4.37</td>
<td>0.03%</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>4.38</td>
<td></td>
</tr>
</tbody>
</table>

isotropic zero-mean Gaussian distribution with covariance of 15 \( mm^2 \mathbf{I}_{3 \times 3} \). Then, we register the data set \( Y \) to \( U \) using the least-mean-squares algorithm proposed by Arun [2], and calculated the TRE at \( r \). As before, this simulation is repeated 100,000 times to numerically estimate the distribution of TRE and its mean-squared value at \( r \). Furthermore, we use (6.41) and (6.92) as well as Fitzpatrick and West formulations to estimate the distribution of TRE and its mean-squared value.

Figure 6.14 shows the estimated distribution of TRE at \( r = [-38.4, -13.7, 97.2]^T \), in \( mm \), using numerical simulations, Fitzpatrick and West algorithm, and the proposed algorithm for one simulation run. Table 6.7 displays the estimated mean-squared values of TRE using these three algorithms as well as the distance among the estimated distributions. As expected, all three algorithms accurately and closely follow each other.

Again, we repeat this simulation 100 times. Table 6.10 shows mean of distance between the distributions estimated by the proposed algorithm and the one generated by numerical simulations, when \( N \) (number of fiducials) is 3, 4, 10, 20, and 50, respectively. The distance between the distributions estimated by the proposed algorithm and Fitzpatrick and West algorithm is zero since both algorithms are exactly
the same when FLE has identical isotropic zero-mean Gaussian distribution at each point of the data sets. The table also displays mean and variance of error difference between estimated root mean-squared values of TRE using the proposed algorithm, Fitzpatrick and West algorithm, and the one generated by numerical simulations. As can be seen from Table 6.10, the proposed algorithm accurately follows the results generated by numerical simulations.

![Figure 6.14: Probability densities of the estimated target registration error at $r$ using numerical simulations, Fitzpatrick and West and the proposed algorithms when the data sets are perturbed by identical isotropic zero-mean Gaussian FLE.](image)

c) FLE has an identical anisotropic zero-mean Gaussian distribution, $\mathcal{N}(0, diag(\sigma_x^2, \sigma_y^2, \sigma_z^2))$, at each point of the data sets:

In this simulation, data set $U$ and target location $r$ are generated as before. To produce data set $Y$, each individual point in $U$ is perturbed by identical anisotropic zero-mean noise with covariance matrix $diag(5 \text{ mm}^2, 15 \text{ mm}^2, 25 \text{ mm}^2)$. As in the
Table 6.7: Estimated mean-squared values of TRE and distance between estimated distributions of TRE at $\mathbf{r}$ using numerical simulations, Fitzpatrick and West and the proposed algorithms when the data sets are perturbed by identical isotropic zero-mean Gaussian FLE.

<table>
<thead>
<tr>
<th>Method</th>
<th>$E<a href="%5Ctext%7Bmm%7D%5E2">\text{TRE}^2</a>$</th>
<th>$d(pdf_1,pdf_2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical simulations</td>
<td>12.71</td>
<td>-</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>12.68</td>
<td>0.02%</td>
</tr>
<tr>
<td>Fitzpatrick and West’s</td>
<td>12.68</td>
<td></td>
</tr>
</tbody>
</table>

First simulation, we use the UKF registration algorithm to register the two data sets, and to calculate TRE at the target location $\mathbf{r}$. As before, this simulation is repeated 100,000 times.

Figure 6.15 shows the estimated distributions of TRE using the proposed algorithm and numerical simulations. It also gives the estimated mean-squared values of TRE and the distance between the estimated distributions of TRE using the proposed algorithm (Equation (6.91)) and numerical simulations listed in Table 6.8. As before, Table 6.10 shows the mean and variance of distance between the two estimated distributions, and the mean and variance of error difference between the root mean-squared values of TRE calculated by the proposed algorithm and numerical simulations, when $N$ (number of fiducials) is 3, 4, 10, 20, and 50, respectively.

As can be seen from Figure 6.15, Tables 6.8 and 6.10, the proposed algorithm’s results closely match those obtained by numerical simulations for the cases where FLE has identical anisotropic zero-mean Gaussian distribution at each point of the data sets.
Figure 6.15: Probability densities of the estimated target registration error at \( r \) using numerical simulations and the proposed algorithm when the data sets are perturbed by identical anisotropic zero-mean Gaussian FLE.

Table 6.8: Estimated mean-squared values of TRE and distance between estimated distributions of TRE at \( r \) using numerical simulations and the proposed algorithm when the data sets are perturbed by identical anisotropic zero-mean Gaussian FLE.

<table>
<thead>
<tr>
<th>Method</th>
<th>( E<a href="mm%5E2">TRE^2</a> )</th>
<th>( d(pdf_1,pdf_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical simulations</td>
<td>13.82</td>
<td>0.08%</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>13.33</td>
<td></td>
</tr>
</tbody>
</table>
d) FLE has an inhomogeneous anisotropic zero-mean Gaussian distribution, \( \mathcal{N}(0, \Lambda_i) \), at each point of the data sets:

In this simulation, data set \( U \) and target location \( r \) are generated as before. To generate data set \( Y \), each point in \( U \) is perturbed by inhomogeneous anisotropic zero-mean Gaussian noise with covariance matrix \( \Sigma_i^\gamma \). \( \Sigma_i^\gamma \) is assumed to be:

\[
\Sigma_i^\gamma = \begin{pmatrix}
(\sigma_x^i)^2 & 0 & 0 \\
0 & (\sigma_y^i)^2 & 0 \\
0 & 0 & (\sigma_z^i)^2
\end{pmatrix},
\]

(6.100)

where \((\sigma_x^i)^2\), \((\sigma_y^i)^2\) and \((\sigma_z^i)^2\) are independently and randomly drawn between 0 and 30 \( mm^2 \). The data set \( Y \) is then registered to \( U \) using the UKF point-based registration algorithm, and TRE at \( r \) is calculated. Again, this procedure is reiterated 100,000 times in order to numerically estimate the distribution of TRE and the mean-squared value of TRE. Equations (6.41) and (6.91) are also utilized to estimate the distribution of TRE and its mean-squared value.

Figure 6.16 shows the estimated distributions of TRE using the proposed algorithm and numerical simulations. Furthermore, the estimated mean-squared value of TRE and the distance between the estimated TRE distributions are displayed in Table 6.9. As before, the mean and the variance of error difference between the computed root mean-squared values of TRE using the proposed and numerical simulations, and the mean distance between the two estimated distributions over 100 trials, when \( N \) (number of fiducials) is 3, 4, 10, 20, and 50, are respectively displayed in Table 6.10.

As evidenced in Tables 6.9 and 6.10, and Figure 6.16, the proposed algorithm provides accurate estimates of the distribution of TRE and its mean-squared value even when FLE has an inhomogeneous anisotropic zero-mean Gaussian distribution.
CHAPTER 6. MEASUREMENT OF THE REGISTRATION ACCURACY

Figure 6.16: Probability densities of the estimated target registration error at \( r \) using numerical simulations and the proposed algorithm when the data sets are perturbed by inhomogeneous anisotropic zero-mean Gaussian FLE at each point of the data sets.

Table 6.9: Estimated mean-squared values of TRE and distance between estimated distributions of TRE at \( r \) using numerical simulations and the proposed algorithm when the data sets are perturbed by inhomogeneous anisotropic zero-mean Gaussian FLE.

<table>
<thead>
<tr>
<th>Method</th>
<th>( \text{E}<a href="%5Ctext%7Bmm%7D%5E2">\text{TRE}^2</a> )</th>
<th>( d(pdf_1, pdf_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical simulations</td>
<td>6.25</td>
<td>0.03%</td>
</tr>
<tr>
<td>Proposed algorithm</td>
<td>6.23</td>
<td></td>
</tr>
</tbody>
</table>
Table 6.10: Mean (mm) and variance (mm\(^2\)) of error difference between the computed root mean squared values of TRE at different target locations using the proposed algorithm and numerical simulations, and the mean distance between the estimated distributions using the proposed algorithm and numerical simulations over 100 trials. N, number of fiducials, is 3, 4, 10, 20, and 50, respectively. “×” means an algorithm is not applicable to calculate the TRE characteristics.

<table>
<thead>
<tr>
<th>N</th>
<th>Simulation</th>
<th>Proposed</th>
<th>Fitzpatrick and West</th>
<th>d(pdf(_1),pdf(_2))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mean</td>
<td>variance</td>
<td>mean</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>-0.09</td>
<td>0.03</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>-0.1</td>
<td>0.03</td>
<td>-0.1</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>-0.2</td>
<td>0.1</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>-0.07</td>
<td>0.01</td>
<td>×</td>
</tr>
<tr>
<td>4</td>
<td>a</td>
<td>-0.02</td>
<td>2×10(^{-3})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>-0.01</td>
<td>3×10(^{-3})</td>
<td>-0.01</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>-0.03</td>
<td>9×10(^{-3})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>-0.02</td>
<td>1×10(^{-3})</td>
<td>×</td>
</tr>
<tr>
<td>10</td>
<td>a</td>
<td>-2×10(^{-3})</td>
<td>6×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>7×10(^{-4})</td>
<td>6×10(^{-5})</td>
<td>7×10(^{-4})</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>3×10(^{-3})</td>
<td>8×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>-1×10(^{-3})</td>
<td>6×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td>20</td>
<td>a</td>
<td>-6×10(^{-4})</td>
<td>1×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>3×10(^{-3})</td>
<td>4×10(^{-5})</td>
<td>3×10(^{-3})</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>-2×10(^{-3})</td>
<td>2×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>4×10(^{-4})</td>
<td>1×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td>50</td>
<td>a</td>
<td>9×10(^{-4})</td>
<td>3×10(^{-6})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>b</td>
<td>-4×10(^{-4})</td>
<td>1×10(^{-5})</td>
<td>-4×10(^{-4})</td>
</tr>
<tr>
<td></td>
<td>c</td>
<td>-1×10(^{-3})</td>
<td>1×10(^{-5})</td>
<td>×</td>
</tr>
<tr>
<td></td>
<td>d</td>
<td>8×10(^{-4})</td>
<td>6×10(^{-6})</td>
<td>×</td>
</tr>
</tbody>
</table>
e) Comparison of the proposed, Sielhorst, Ma, Wiles, and Fitzpatrick and West algorithms:

In this simulation, as before, data set $U$ is generated by drawing 10 fiducial points uniformly within a cube with the sides of $\pm 100 \text{ mm}$. Also, $r$, the target location, is selected randomly from a cube of sides $\pm 200 \text{ mm}$. For each data set $U$, we generated two data sets $Y_1$ and $Y_2$ contaminated by identical isotropic zero-mean Gaussian noise and inhomogeneous anisotropic zero-mean Gaussian noise, respectively, by perturbing independently the $x$, $y$, and $z$ components of each point in $U$ by a zero-mean Gaussian random variable which models FLE along each orthogonal axis. The distribution of FLE is considered to be $\mathcal{N}(0, \sigma^2 I)$, and $\mathcal{N}(0, \text{diag}((\sigma_x^i)^2, (\sigma_y^i)^2, (\sigma_z^i)^2))$ for the isotropic and anisotropic cases, respectively. $\sigma^2$ is assumed to be $10 \text{ mm}^2$ and $(\sigma_x^i)^2$, $(\sigma_y^i)^2$, and $(\sigma_z^i)^2$ are randomly chosen from a uniform distribution with sides of $0$ and $20 \text{ mm}^2$ for each point in data set $U$. In this way, we assume FLE has an identical isotropic zero-mean Gaussian distribution and inhomogeneous anisotropic zero-mean Gaussian distribution for each point in the data set, respectively. Then, $U$ is registered to $Y_1$ and $Y_2$ by using the UKF point-based registration algorithm, and TRE is measured at $r$.

The distribution of TRE is numerically estimated by using repeated perturbation and registration steps 100,000 times. Also, the proposed, Sielhorst [114], Ma [71], Wiles [133], and Fitzpatrick and West [36] algorithms are used to compute the distribution of TRE. Figures 6.17 and 6.18 display the computed probability density functions of TRE at $r = [184, 91, -35]^T$, in $\text{mm}$, using numerical simulations and the other algorithms when FLE has an inhomogeneous anisotropic, and identical isotropic zero-mean Gaussian distribution, respectively.
CHAPTER 6. MEASUREMENT OF THE REGISTRATION ACCURACY

Figure 6.17: Probability density of TRE at $r$, when FLE has an inhomogeneous anisotropic zero-mean Gaussian distribution. Please note that the proposed, Sielhorst, and Ma algorithms’ result is overlaid on that of numerical simulations. Also, the results of the Fitzpatrick and West algorithm and Wiles algorithm are overlaid on each other.

Figure 6.18: Probability density of TRE at $r$, when FLE has identical isotropic zero-mean Gaussian noise. Please note that all the curves generated from all the algorithms, exactly follow each other when FLE has an identical isotropic zero-mean Gaussian distribution.
Figure 6.17 illustrates that, if FLE has an inhomogeneous and anisotropic zero-mean Gaussian noise, Sielhorst [114], and Ma [71] algorithms converge to our proposed ML solution. Also, as shown in Figure 6.18, all algorithms including Fitzpatrick and West [36] and Wiles [133] derive the same distribution of TRE at a target location when FLE has an identical isotropic zero-mean Gaussian noise.

6.3 Summary and Discussion

We, first, employed the UKF registration algorithm proposed in Chapter 4, to estimate the distribution of TRE and the mean-squared value of TRE when FLE has an identical isotropic zero-mean Gaussian distribution. Based on the variances exploited by the UKF registration algorithm, (6.41) and (6.42) were derived to estimate the TRE distribution and its mean-squared value, respectively. Figures 6.1 and 6.3 and Tables 6.1 and 6.2 illustrated that the proposed derivations and formulae estimate the distribution of TRE and its mean-squared value as accurately as those obtained by Fitzpatrick and West [36] and numerical simulations.

Furthermore, Tables 6.3 and 6.4 displayed the close agreement of the proposed and Fitzpatrick and West algorithms in estimation of the distribution of TRE and its mean-squared value. The main advantage of the proposed algorithm is that, unlike the Fitzpatrick and West algorithm, it does not need the Monte-Carlo simulation to estimate the distribution of TRE.

The loci of the points with the same mean-squared value of TRE were also derived in (6.48) and, as displayed in Figures 6.5 and 6.6, were shown to be ellipsoids. Finally, we used the proposed derivations to generate a color map displaying the mean-squared value of TRE at every point of the registration model after performing
registration, as shown in Figures 6.8, 6.10, and 6.12. We hope this map gives surgeons more reliable information about the accuracy of the performed registration at their intended target locations.

We then used ML algorithm to estimate the distribution of TRE and its mean-squared value when FLE has an arbitrary distribution at each point of the registering data sets. Based on the ML algorithm, we evidenced that the distribution of target registration error and its mean-squared value can be estimated using the variance of the registration parameters (rotation and translation parameters) when the registering data sets are perturbed by fiducial localization error (FLE) of any distribution. The variance of registration parameters were estimated by using the ML algorithm, based on the assumption that the moving and fixed data sets are modeled by (6.1) \((i.e., \text{the two data sets are nearly registered together})\).

When data sets are perturbed by FLE with any variant of independent zero-mean Gaussian distribution, we showed that the ML algorithm can estimate the registration parameters and their variances using (6.55), (6.56), (6.62), (6.63), (6.64), and (6.65), respectively. The estimated variance of registration parameters was then utilized in (6.80), (6.81), (6.82), and (6.83) to derive the covariance matrix of TRE at a target location \( \mathbf{r}, \Sigma_{er} \). Subsequently, in (6.41), the eigenvalues of \( \Sigma_{er} \) were used to estimate the distribution of TRE at \( \mathbf{r} \).

Furthermore, the variance of registration parameters was utilized in (6.91) to estimate the mean-squared value of TRE.

When FLE has identical isotropic zero-mean Gaussian noise at each point of the data sets, the proposed formulations were analytically shown to match the ones presented by Fitzpatrick and West [36]. In addition, we mathematically displayed that
if FLE has an inhomogeneous anisotropic zero-mean Gaussian distribution at each point of the data sets, the proposed formulations match the ones presented by Sielhorst [114], and Ma [71], respectively.

To also numerically verify the performance of the proposed algorithm in the presence of FLE with a different form of Gaussian distribution at each point of the registering data sets, five simulations were performed. The latter not only numerically measured the mean-squared value of TRE and its distribution, but also compared them with the proposed formulations’ results when FLE has identical or inhomogeneous, isotropic or anisotropic zero-mean Gaussian noise at each point of the data sets. Simulation results verified that the proposed algorithm correctly estimates the mean-squared value of TRE and the distribution of TRE for any Gaussian distributions that model FLE at each point of the data sets. Figures 6.13, 6.14, 6.15, and 6.16 illustrated that the proposed algorithm accurately estimates the distribution of TRE in the presence of different forms of FLE distributions. Tables 6.6, 6.7, 6.8, 6.9, and 6.10 displayed the close agreement of the proposed algorithm with numerical simulations in estimating the mean-squared value of TRE at the target point \( r \) when FLE has an inhomogeneous isotropic zero-mean Gaussian distribution; identical isotropic zero-mean Gaussian distribution; identical anisotropic zero-mean Gaussian distribution; and inhomogeneous anisotropic Gaussian distribution, respectively, at each point of the data sets. An interesting result transpires from Simulations (a), (b), and (d), respectively. While in those simulations, the average power of the noise is the same (15 \( mm^2 \)) along each orthogonal axis, the computed root mean-squared value of TRE at \( r \) is different. This finding signifies that if FLE has an inhomogeneous isotropic zero-mean Gaussian distribution, then the average of the distribution for all
the points in the data sets, cannot be employed as an identical isotropic zero-mean Gaussian distribution within the Fitzpatrick and West algorithm to estimate the TRE characteristics.

Finally, Figure 6.17 showed that if FLE has an inhomogeneous anisotropic zero-mean Gaussian noise, then Sielhorst [114] and Ma [71] algorithms converge to the ML solutions. Furthermore, as shown in Figure 6.18, all algorithms including Fitzpatrick and West [36] and Wiles [133] derive the same distribution of TRE at the target location when FLE has an identical isotropic zero-mean Gaussian noise.

Before finishing this chapter, we would like to emphasize that the derived covariance matrices $\Sigma_{x_t}$, $\Sigma_{x_t x_\theta}$, and $\Sigma_{x_\theta}$, (Equations (6.62), (6.63), and (6.64)), used in estimating the TRE distribution and its mean-squared value are valid as long as FLE can be modeled by a Gaussian distribution. Should the distribution of FLE be non-Gaussian, then that distribution should be used in (6.51) to calculate the new log likelihood function and its first and second derivatives from (6.53), (6.54), (6.59), (6.60), and (6.61), respectively. The second derivatives of the new log-likelihood function can then be employed in (6.65) to estimate $\Sigma$. After estimating $\Sigma$, the remainder of the proposed algorithm may be utilized to compute the TRE characteristics.
Chapter 7

Conclusions

In this dissertation, point- and surface-based rigid body registration algorithms were discussed, and various new registration methods based on the UKF algorithm were proposed. The UKF algorithm was used as the basis for the registration algorithms since it is a weighted least-mean-squares algorithm that can increase the accuracy and reliability of the performed registration in the presence of an arbitrary Gaussian noise. It not only estimates the registration parameters but also computes the variance of the estimated registration parameters. Those variances were then used to compute the accuracy of the performed registration.

In Chapter 2 of this thesis, we first explained point- and surface-based rigid body registration problems in the literature. To prepare a common basis and introduce our new registration algorithms, Kalman Filter and its extensions, EKF and UKF, were discussed in Chapter 3.

In Chapter 4, a new pairwise point-based registration algorithm based on UKF was presented. It was numerically shown that the proposed algorithm can iteratively and accurately estimate the registration parameters and their variances, in contrast
to other methods in the literature, such as Arun [2], Umeyama [126] and Pennec [98] registration algorithms.

The proposed algorithm was then extended to perform surface-based registration between two data sets (point sets with unknown correspondences) which were assumed to be roughly aligned. By means of numerical simulations, it was shown that the proposed surface-based registration algorithm can accurately and robustly register two surfaces. Also, the proposed algorithm was compared to the ICP registration algorithm. It was verified that the proposed surface-based registration algorithm (due to the UKF property that can accurately accounts for the noise information in the system) was more robust, accurate, and less sensitive to the initial alignment error than the ICP registration algorithm.

In Chapter 5, the proposed pair-wise registration methods were extended to multi-body registration algorithms, and two new UKF multi-body registration techniques were introduced. In the first technique, a set of data sets with known correspondences was accurately registered in a common coordinate frame. It was numerically shown that the proposed algorithm registers the data sets more accurately than other common algorithms in the literature such as Pennec’s [97].

The proposed multi-body point-based registration algorithm was then extended to perform multi-body surface-based registration among surfaces which did not rely on pre-existing correspondences, although the surfaces must still be roughly aligned. The proposed multi-body surface-based registration algorithm was then assessed on a clinical application which involved automatic alignment of a set of fractured bones to a solid bone template. It was shown that the proposed algorithm can be used to accurately register multiple femur and pelvic bone fractures back into their correct
healing positions.

Finally, in Chapter 6, the variances computed by the proposed UKF registration algorithm, were used to derive (6.41) and (6.42) that approximate the distribution of TRE and mean-squared value of TRE at any desired target location when FLE has an isotropic and identical zero-mean Gaussian distribution. It was numerically shown that the proposed formula accurately computes the distribution of TRE in a single simulation run, and closely follows the Fitzpatrick and West’s results derived by Monte Carlo simulations [36]. Then, using the Maximum Likelihood algorithm, we extended our derivations and proposed a new solution to estimate the characteristics of TRE (the distribution and mean-squared value of TRE) when FLE has an arbitrary distribution. By formulating (6.59), (6.60), (6.61), and (6.65), we derived, for the first time, expressions for the mean-squared value of TRE and the distribution of TRE in the presence of different types of fiducial localization error (FLE) distribution at each point of the data sets. That our proposed algorithm and derivations closely agree with the performed numerical simulations demonstrates its high level of accuracy in estimating the characteristics of TRE in the presence of an arbitrary FLE.

7.1 Future Work

This dissertation has made major contributions to understanding rigid-body point- and surface-based registration and estimating the accuracy of the performed registration; but, it has left many issues to be answered and investigated. Some future research work are presented here:

1) It would be interesting to thoroughly compare the performance of the proposed
UKF pairwise point-based registration algorithm with the other methods in the literature, such as Balachandran [3] and Matei [74] algorithms which register two data sets in the presence of inhomogeneous and anisotropic Gaussian noise. Furthermore, examining the performance of the proposed algorithms in the presence of non-Gaussian FLE is another area for further investigation.

2) We only compared the proposed UKF pairwise surface-based registration algorithm with the standard ICP registration algorithm. It would be of interest to compare our algorithm with those in Grenger [43] and Ranjaparan [103] which claim to have better registration results than ICP.

Furthermore, in our proposed UKF pairwise surface-based registration algorithms, it was assumed that data sets are roughly aligned such that the closest points among them are a good estimate of the corresponding points. It would be interesting to ease this limitation such that the data sets would not required to be preliminarily aligned before performing the registration. One potential way to eliminate this constraint is to use local point descriptors, as they can be employed to automatically find a set of corresponding points among the data sets and to roughly align them. There are many local point descriptors defined in the literature which can be combined with the proposed registration algorithms to automatically register data sets without any preliminarily alignments.

3) Another issue to investigate is the performance of the proposed algorithms in estimating the distribution of TRE and its mean-squared value in the presence of non-Gaussian FLE. We only verified the performance of the proposed derivations when FLE has an arbitrary Gaussian distribution. It would be of interest to numerically calculate the variance of the transformation parameters when FLE has a
CHAPTER 7. CONCLUSIONS

non-Gaussian distribution by using the Maximum Likelihood function and then, estimate the distribution of TRE at a target location.

4) Rigid-body point- and surface-based registration are types of registration modality that are important in computer-assisted and image-guided surgeries. Image (intensity)-based registration, which involves matching a medical image to a computer model or a medical image is another important method of registration. Understanding the registration error associated with image-based registration appears to be a very challenging subject of research, in part because it requires an understanding of the physics and uncertainties of the image formation. It is our hope that the proposed algorithms estimating the accuracy of the performed registration could be used to estimate the accuracy of performed image-based registration in the near future.
Bibliography


Appendix A

Statistical Anatomical Atlas Generation

In this section, we explain how to use the PCA approach to generate the anatomical atlas from many three-dimensional bone surface models. The bone models are generated from Computed Tomography (CT) images of a specific bony anatomy over a population. One of the models is randomly selected as a template, and all other models are registered to the template using similarity transformations to bring them in the same coordinate space as the template. Let us represent the template with $y_0$ as a $1 \times 3N_p$ vector, where $N_p$ is the number of the points in the model. Also, let $y_i$, $i = 1, ..., N_m - 1$, represent the points in other models which are transformed to the template coordinate frame ($N_m$ is the total number of models). Now, let us define the mean shape as follows:

$$\mathbf{m} = \frac{1}{N_m} \sum_{i=0}^{N_m-1} y_i.$$  \hspace{1cm} (A.1)
Using the mean shape, the coordinates of the points in all models can be generated as follows:

\[ y_i = m + \sum_{k=1}^{N_m-1} \alpha_k u_k, \quad \alpha_k = u_k^T(y_i - m), \quad (A.2) \]

where \( u_k \) is the \( k \)’th eigenvector of deformation matrix \( D \) defined as follows:

\[ D = [y_0 - m, y_1 - m, ..., y_{N_m} - m]. \quad (A.3) \]

If singular values of matrix \( D \) are represented by \( \lambda_i \), then a new instance of the atlas can be generated from

\[ y_c = m + \sum_{i=1}^{N_m-1} c_i u_i, \quad (A.4) \]

where \( c_i \) is randomly chosen from a uniform distribution in the range of \([-\sqrt{\frac{3\lambda_i}{N_m}}, \sqrt{\frac{3\lambda_i}{N_m}}]\).
Appendix B

Local Point Descriptor

A number of local point descriptors which are invariant to the linear transformations have been introduced in the literature to find the potential correspondences among point data sets. Nearly all the introduced local point descriptors are based on discrete points. Here, we propose a new local point descriptor which is based on continuous functions, fitted to a group of neighboring points. In what follows, we explain the proposed descriptor used to find the potential matches between two data sets, \( y_0 \) and \( y_1 \):

1) The algorithm first randomly selects a point \( p_0 \) from \( y_0 \). It then finds points which are within a certain distance \( d \) from \( p_0 \). Let \( P = \{p_0, p_1, ..., p_n\} \) be the set of chosen points from \( y_0 \).

2) The translation invariance is achieved by translating the centroid of \( P \), \( c \), to the origin.

3) The rotation invariance is accomplished by rotating all the points in \( P \) with a rotation matrix \( R \), which aligns the point cloud along its eigenvectors.

4) The reflection invariance is obtained by multiplying the coordinates of points
P with matrix $F = \text{diag}(\text{sign}(f_x), \text{sign}(f_y), \text{sign}(f_z))$, where $f_x$, $f_y$ and $f_z$ are the signed mean square of $P$ along $x$, $y$, and $z$ axes, respectively.

5) Finally, the scale invariance is provided by scaling factor $s = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$, where $\sigma_x^2$, $\sigma_y^2$ and $\sigma_z^2$ are the variances of $P$ along $x$, $y$ and $z$, respectively.

The above procedure brings the points in $P$ to the canonical coordinate frame as follows:

$$\tilde{P} = \frac{1}{s} \times F \times R \times (P - \underbrace{[c,c,...,c]}_{n \times 1}).$$  \hspace{0.5cm} (B.1)

Then, a continuous surface function is fitted to the points in $\tilde{P}$ such that the mean square error of the fitness quality is minimized. The order of the function depends on the complexity of the surface. In our study, we used a third-order function as

$$f(x, y, z) = a_1 x^3 + a_2 y^3 + a_3 z^3 + a_4 x^2 y + a_5 x^2 z + a_6 y^2 x + a_7 y^2 z + a_8 z^2 x + a_9 z^2 y + a_{10} xyz + a_{11} x^2 + a_{12} y^2 + a_{13} z^2 + a_{14} xy + a_{15} xz + a_{16} yz + a_{17} x + a_{18} y + a_{19} z + 1,$$ \hspace{0.5cm} (B.2)

where $a_{p_0} = [a_1, ..., a_{19}]^T$ can be easily computed using a closed-form solution minimizing the mean-squares of the fitness error. Equation (B.2) should be satisfied for each point $p_i = [x_{p_i}, y_{p_i}, z_{p_i}]^T$ in $\tilde{P}$

$$f(x_{p_i}, y_{p_i}, z_{p_i}) = -1, \quad p_i \in \tilde{P}.$$ \hspace{0.5cm} (B.3)

Equation (B.3) can then be written in a matrix format as $M \times a_{p_0} = b$, where $M$ is a $n \times 19$ matrix in terms of $p_i$ and $b = [-1, ..., -1]^T_{1 \times 19}$. Therefore, $a_{p_0}$ can be calculated as

$$a_{p_0} = \text{inv}(M^T \times M) \times M^T \times b.$$ \hspace{0.5cm} (B.4)

We consider $a_{p_0}$ as an attribute vector for the point $p_0$. To find the potential matching
points of \( p_0 \) in \( y_1 \), the attribute vectors for all the points in \( y_1 \) are calculated and compared with \( a_{p_0} \). A simple similarity measure such as normalized cross-correlation can be used to find the potential matches of \( p_0 \) in \( y_1 \).
Appendix C

Proof of Cramer-Rao inequality

The proof is a simple application of the Schwarz inequality. Let us define vector $\mathbf{x}$ including the registration parameters, as $\mathbf{x} = [t_x, t_y, t_z, \theta_z, \theta_y, \theta_x]^T$. Since it is assumed that the estimated registration parameters are unbiased, one can write

$$E[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}] \triangleq \int_{-\infty}^{+\infty} P(\mathbf{Y}|\mathbf{U}, \mathbf{x})[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}]d\mathbf{Y} = 0. \quad (C.1)$$

Differentiating both sides with respect to $\mathbf{a}$, we have:

$$\frac{\partial}{\partial \mathbf{x}} \int_{-\infty}^{+\infty} P(\mathbf{Y}|\mathbf{U}, \mathbf{x})[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}]d\mathbf{Y} = \int_{-\infty}^{+\infty} \frac{\partial}{\partial \mathbf{x}} \{P(\mathbf{Y}|\mathbf{U}, \mathbf{x})[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}]\}d\mathbf{Y} = 0, \quad (C.2)$$

Then

$$-\int_{-\infty}^{+\infty} P(\mathbf{Y}|\mathbf{U}, \mathbf{x})d\mathbf{Y} + \int_{-\infty}^{+\infty} \frac{\partial P(\mathbf{Y}|\mathbf{U}, \mathbf{x})}{\partial \mathbf{x}}[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}]d\mathbf{Y} = 0. \quad (C.3)$$

The first integral is 1. Now, since

$$\frac{\partial P(\mathbf{Y}|\mathbf{U}, \mathbf{x})}{\partial \mathbf{x}} = \frac{\partial \ln P(\mathbf{Y}|\mathbf{U}, \mathbf{x})}{\partial \mathbf{x}} P(\mathbf{Y}|\mathbf{U}, \mathbf{x}), \quad (C.4)$$

one can substitute (C.4) into (C.3) to have

$$\int_{-\infty}^{+\infty} \frac{\partial \ln P(\mathbf{Y}|\mathbf{U}, \mathbf{x})}{\partial \mathbf{x}} P(\mathbf{Y}|\mathbf{U}, \mathbf{x})[\hat{\mathbf{x}}(\mathbf{Y}, \mathbf{U}) - \mathbf{x}]d\mathbf{Y} = 1. \quad (C.5)$$
By rewriting (C.5), we have
\[
\int_{-\infty}^{+\infty} \left( \frac{\partial \ln P(Y|U, x)}{\partial x} \sqrt{P(Y|U, x)} \times \left[ \sqrt{P(Y|U, x)} [\hat{x}(Y, U) - x] \right] \right) dY = 1,
\]
(C.6)
and using the Schwarz inequality, we have
\[
\left( \int_{-\infty}^{+\infty} \left[ \frac{\partial \ln P(Y|U, x)}{\partial x} \right]^2 P(Y|U, x) dY \right) \times \left( \int_{-\infty}^{+\infty} [\hat{x}(Y, U) - x]^2 P(Y|U, x) dY \right) \geq 1,
\]
(C.7)
where we recall from the derivation of the Schwarz inequality that equality holds if, and only if
\[
\hat{x}_i(Y, U) - x_i = \sum_{j=1}^{6} k_{ij}(x) \frac{\partial P(Y|U, x)}{\partial x_j},
\]
(C.8)
for all \(Y, U, \) and \(x_i\). \(k_{ij}\) is an arbitrary function. It can be easily seen that the two terms of the left side of (C.7) are expectations, therefore
\[
E\{[\hat{x}(Y, U) - x]^2\} \geq \left( E \left[ \frac{\partial \ln P(Y|U, x)}{\partial x} \right]^2 \right)^{-1}.
\]
(C.9)
To prove Equation (6.65), we observe
\[
\int_{-\infty}^{+\infty} P(Y|U, x) dY = 1.
\]
(C.10)
Differentiating with respect to \(a\), we have
\[
\int_{-\infty}^{+\infty} \frac{\partial P(Y|U, x)}{\partial x} dY = \int_{-\infty}^{+\infty} \frac{\partial \ln P(Y|U, x)}{\partial x} P(Y|U, x) dY = 0.
\]
(C.11)
Differentiating again with respect to \(a\) and applying (C.4), we have
\[
\int_{-\infty}^{+\infty} \frac{\partial^2 \ln P(Y|U, x)}{\partial x^2} P(Y|U, x) dY + \int_{-\infty}^{+\infty} \left[ \frac{\partial \ln P(Y|U, x)}{\partial x} \right]^2 P(Y|U, x) dY = 0,
\]
or
\[
E\left[ \frac{\partial^2 \ln P(Y|U, x)}{\partial x^2} \right] = -E\left[ \left( \frac{\partial \ln P(Y|U, x)}{\partial x} \right)^2 \right],
\]
(C.12)
which together with (C.9) gives
\[
E\{[\hat{x}(Y, U) - x]^2\} \geq -\left(E\left[\partial^2 \ln P(Y|U, x) \right] \right)^{-1}
\]  \hspace{1cm} (C.13)

Equation (C.13) completes the proof.

It is very easy to verify that since, in our case, the estimated rotation and translation parameters can be written in shape of (C.8), the Equality in (C.13) holds, and it leads to (6.65).